Solutions to Darwiche's Modeling and Reasoning with Bayesian Networks

Aayush Bajaj

2026-09-07T09:00:00+10:00

Contents

1 Propositional Logic **3**
1.1 Exercises 2.1–2.7 . . . 4
1.2 Exercises 2.8–2.13 . . . 6

2 Probability Calculus **11**
2.1 Exercises 3.1–3.7 . . . 12
2.2 Exercises 3.8–3.14 . . . 15
2.3 Exercises 3.15–3.21 . . . 20
2.4 Exercises 3.22–3.27 . . . 23

3 Bayesian Networks **28**
3.1 Exercises 4.1–4.7 . . . 29
3.2 Exercises 4.8–4.14 . . . 34
3.3 Exercises 4.15–4.21 . . . 39
3.4 Exercises 4.22–4.25 . . . 44

4 Building Bayesian Networks **48**
4.1 Exercises 5.1–5.7 . . . 49
4.2 Exercises 5.8–5.14 . . . 59
4.3 Exercises 5.15–5.16 . . . 66

5 Inference by Variable Elimination **68**
5.1 Exercises 6.1–6.7 . . . 69
5.2 Exercises 6.8–6.14 . . . 74
5.3 Exercises 6.15–6.18 . . . 78

6 Inference by Factor Elimination **81**
6.1 Exercises 7.1–7.7 . . . 82
6.2 Exercises 7.8–7.14 . . . 90
6.3 Exercises 7.15–7.15 . . . 95

7 Inference by Conditioning **97**
7.1 Exercises 8.1–8.7 . . . 98
7.2 Exercises 8.8–8.14 . . . 105

8 Models for Graph Decomposition **114**
8.1 Exercises 9.1–9.7 . . . 115
8.2 Exercises 9.8–9.14 . . . 122
8.3 Exercises 9.15–9.21 . . . 126

8.4 Exercises 9.22–9.28 . . . 132

9 Most Likely Instantiations **139**
9.1 Exercises 10.1–10.7 . . . 140
9.2 Exercises 10.8–10.14 . . . 147
9.3 Exercises 10.15–10.21 . . . 153
9.4 Exercises 10.22–10.22 . . . 159

10 The Complexity of Probabilistic Inference **161**
10.1 Exercises 11.1–11.7 . . . 162
10.2 Exercises 11.8–11.10 . . . 170

11 Compiling Bayesian Networks **174**
11.1 Exercises 12.1–12.7 . . . 175
11.2 Exercises 12.8–12.14 . . . 182
11.3 Exercises 12.15–12.20 . . . 189

12 Inference with Local Structure **194**
12.1 Exercises 13.1–13.7 . . . 195
12.2 Exercises 13.8–13.14 . . . 202
12.3 Exercises 13.15–13.21 . . . 210
12.4 Exercises 13.22–13.22 . . . 217

13 Approximate Inference by Belief Propagation **219**
13.1 Exercises 14.1–14.7 . . . 220
13.2 Exercises 14.8–14.14 . . . 229
13.3 Exercises 14.15–14.20 . . . 237

14 Approximate Inference by Stochastic Sampling **245**
14.1 Exercises 15.1–15.7 . . . 246
14.2 Exercises 15.8–15.14 . . . 250
14.3 Exercises 15.15–15.21 . . . 255
14.4 Exercises 15.22–15.28 . . . 259
14.5 Exercises 15.29–15.31 . . . 266

15 Sensitivity Analysis **269**
15.1 Exercises 16.1–16.7 . . . 270
15.2 Exercises 16.8–16.14 . . . 274
15.3 Exercises 16.15–16.17 . . . 281

16 Learning: The Maximum Likelihood Approach **287**
16.1 Exercises 17.1–17.7 . . . 288
16.2 Exercises 17.8–17.14 . . . 293
16.3 Exercises 17.15–17.20 . . . 300

17 Learning: The Bayesian Approach **307**
17.1 Exercises 18.1–18.7 . . . 308
17.2 Exercises 18.8–18.14 . . . 314
17.3 Exercises 18.15–18.21 . . . 319
17.4 Exercises 18.22–18.24 . . . 326

Solutions to every exercise in Adnan Darwiche's *Modeling and Reasoning with Bayesian Networks* (Cambridge, 2009) — 342 exercises across chapters 2–18, from propositional logic and probability calculus through exact and approximate inference to parameter and structure learning. The book is the COMP9418 reference text and is filed at Modeling and Reasoning with Bayesian Networks.

1 Propositional Logic

Contents

1.1 Exercises 2.1–2.7 4
1.2 Exercises 2.8–2.13 6

1.1 Exercises 2.1–2.7

Problem 2.1. Show that the following sentences are consistent by identifying a world that satisfies each sentence:
(a) $(A \Rightarrow B) \wedge (A \Rightarrow \neg B)$.
(b) $(A \vee B) \Rightarrow (\neg A \wedge \neg B)$.

Solution. Both sentences are satisfied by the world ω with $\omega(A) = \mathsf{false}$ and $\omega(B) = \mathsf{false}$.
(a) Since $\omega \models \neg A$, both $\neg A \vee B$ and $\neg A \vee \neg B$ hold at ω, hence so does their conjunction; here $\alpha \Rightarrow \beta$ abbreviates $\neg\alpha \vee \beta$ (Table 2.2). Distributing and using that $B \wedge \neg B$ is inconsistent (Table 2.2),

$$(A \Rightarrow B) \wedge (A \Rightarrow \neg B) \;=\; \neg A \vee (B \wedge \neg B) \;=\; \neg A,$$

the form Exercise 2.3(b) reuses; its models are the two worlds with A false.
(b) At ω the antecedent $A \vee B$ is false, so the implication holds. Writing $\gamma = A \vee B$, de Morgan turns the sentence into $\gamma \Rightarrow \neg\gamma = \neg\gamma \vee \neg\gamma = \neg A \wedge \neg B$, whose unique model is ω.
The truth table confirms both columns; a sentence is satisfied at the worlds carrying "true" in its column.

A	B	(a) $(A \Rightarrow B) \wedge (A \Rightarrow \neg B)$	(b) $(A \vee B) \Rightarrow (\neg A \wedge \neg B)$
true	true	false	false
true	false	false	false
false	true	true	false
false	false	true	true

Problem 2.2. Which of the following sentences are valid? If a sentence is not valid, identify a world that does not satisfy the sentence.
(a) $(A \wedge (A \Rightarrow B)) \Rightarrow B$.
(b) $(A \wedge B) \vee (A \wedge \neg B)$.
(c) $(A \Rightarrow B) \Rightarrow (\neg B \Rightarrow \neg A)$.

Solution. (a) Valid; (b) not valid, falsified at $\omega(A) = \mathsf{false}$, $\omega(B) = \mathsf{false}$; (c) valid. Throughout $\alpha \Rightarrow \beta$ is $\neg\alpha \vee \beta$, and validity means $\mathrm{Mods}(\alpha) = \Omega$ (Section 2.3.2).
(a) is modus ponens: if some ω falsified it then $\omega \models A$ and $\omega \models \neg A \vee B$ while $\omega \not\models B$, so both disjuncts of $\neg A \vee B$ fail at ω — impossible.
(b) By distribution and the validity of $B \vee \neg B$ (Table 2.2),

$$(A \wedge B) \vee (A \wedge \neg B) \;=\; A \wedge (B \vee \neg B) \;=\; A,$$

so the sentence fails at exactly the worlds setting A to false, the exhibited world among them.
(c) Table 2.2 lists the contrapositive $\neg\beta \Rightarrow \neg\alpha$ as equivalent to $\alpha \Rightarrow \beta$, so antecedent and consequent have the same truth value at every world and the implication is never false.
The truth table records all three verdicts:

A	B	(a)	(b)	(c)
true	true	true	true	true
true	false	true	true	true
false	true	true	false	true
false	false	true	false	true

Problem 2.3. Which of the following pairs of sentences are equivalent? If a pair of sentences is not equivalent, identify a world at which they disagree (one of them holds but the other does not).
(a) $A \Rightarrow B$ and $B \Rightarrow A$.
(b) $(A \Rightarrow B) \wedge (A \Rightarrow \neg B)$ and $\neg A$.
(c) $\neg A \Rightarrow \neg B$ and $(A \vee \neg B \vee C) \wedge (A \vee \neg B \vee \neg C)$.

Solution. (a) Not equivalent — they disagree at $\omega(A) = \mathsf{false}$, $\omega(B) = \mathsf{true}$; (b) and (c) are equivalent. Equivalence means equality of model sets (Section 2.3.3).
(a) At that world $A \Rightarrow B = \neg A \vee B$ holds via $\neg A$, while $B \Rightarrow A = \neg B \vee A$ has both disjuncts false.
(b) By Exercise 2.1(a), $(A \Rightarrow B) \wedge (A \Rightarrow \neg B) = \neg A$, and equivalence-preserving rewriting leaves the model set unchanged.
(c) Both sides reduce to $A \vee \neg B$. On the left, $\neg A \Rightarrow \neg B = \neg\neg A \vee \neg B = A \vee \neg B$; on the right, distribution (Table 2.2) with $\alpha = A \vee \neg B$, $\beta = C$, $\gamma = \neg C$ gives

$$\begin{aligned}(A \vee \neg B \vee C) \wedge (A \vee \neg B \vee \neg C) &= (A \vee \neg B) \vee (C \wedge \neg C) \\ &= A \vee \neg B.\end{aligned}$$

The right sentence therefore does not depend on $\omega(C)$, which is what makes the comparison over the eight worlds below legitimate.

A	B	C	$\neg A \Rightarrow \neg B$	$(A \vee \neg B \vee C) \wedge (A \vee \neg B \vee \neg C)$
true	true	true	true	true
true	true	false	true	true
true	false	true	true	true
true	false	false	true	true
false	true	true	false	false
false	true	false	false	false
false	false	true	true	true
false	false	false	true	true

Problem 2.4. For each of the following pairs of sentences, decide whether the first sentence implies the second. If the implication does not hold, identify a world at which the first sentence is true but the second is not.
(a) $(A \Rightarrow B) \wedge \neg B$ and A.
(b) $(A \vee \neg B) \wedge B$ and A.
(c) $(A \vee B) \wedge (A \vee \neg B)$ and A.

Solution. (a) The implication fails, at $\omega(A) = \mathsf{false}$, $\omega(B) = \mathsf{false}$; (b) and (c) hold. Here $\alpha \models \beta$ means $\mathrm{Mods}(\alpha) \subseteq \mathrm{Mods}(\beta)$ (Section 2.3.3), and β is the atomic sentence A in all three parts.
(a) At the exhibited world $\neg A \vee B$ holds via $\neg A$ and $\neg B$ holds, yet $\omega \not\models A$. Distribution and the inconsistency of $B \wedge \neg B$ (Table 2.2) give

$$(A \Rightarrow B) \wedge \neg B \;=\; (\neg A \wedge \neg B) \vee (B \wedge \neg B) \;=\; \neg A \wedge \neg B,$$

so the first sentence entails $\neg A$ instead — modus tollens.
(b) $(A \vee \neg B) \wedge B = (A \wedge B) \vee (\neg B \wedge B) = A \wedge B$, and $\mathrm{Mods}(A \wedge B) = \mathrm{Mods}(A) \cap \mathrm{Mods}(B) \subseteq \mathrm{Mods}(A)$.
(c) $(A \vee B) \wedge (A \vee \neg B) = A \vee (B \wedge \neg B) = A$, so the implication holds, in both directions.
Over the four worlds:

A	B	(a) $(A \Rightarrow B) \wedge \neg B$	(b) $(A \vee \neg B) \wedge B$	(c) $(A \vee B) \wedge (A \vee \neg B)$	A
true	true	false	true	true	true
true	false	false	false	true	true
false	true	false	false	false	false
false	false	true	false	false	false

Problem 2.5. Which of the following pairs of sentences are mutually exclusive? Which are exhaustive? If a pair of sentences is not mutually exclusive, identify a world at which they both hold. If a pair of sentences is not exhaustive, identify a world at which neither holds.
(a) $A \vee B$ and $\neg A \vee \neg B$.
(b) $A \vee B$ and $\neg A \wedge \neg B$.

(c) A and $(\neg A \vee B) \wedge (\neg A \vee \neg B)$.

Solution. (a) Not mutually exclusive — both hold at $\omega(A) = \mathsf{true}$, $\omega(B) = \mathsf{false}$ — but exhaustive; (b) and (c) are both mutually exclusive and exhaustive. By Section 2.3.3 these ask whether $\mathrm{Mods}(\alpha) \cap \mathrm{Mods}(\beta) = \emptyset$ and whether $\mathrm{Mods}(\alpha) \cup \mathrm{Mods}(\beta) = \Omega$.
(a) At the exhibited world $A \vee B$ holds via A and $\neg A \vee \neg B$ via $\neg B$. The first sentence excludes only the world with A and B both false, the second only the world with both true, so no world is excluded by both: the pair is exhaustive. (Equivalently, the disjunction of the two has A and $\neg A$ among its disjuncts.)
(b) By de Morgan (Table 2.2) the second sentence is $\neg(A \vee B)$, the negation of the first, and $\mathrm{Mods}(\neg\gamma) = \overline{\mathrm{Mods}(\gamma)}$; a set and its complement are disjoint and cover Ω.
(c) Distribution and the inconsistency of $B \wedge \neg B$ (Table 2.2) give $(\neg A \vee B) \wedge (\neg A \vee \neg B) = \neg A$, so the pair is A and $\neg A$, again complementary.
In the table below, mutual exclusivity of a pair means no row carries "true" in both of its columns, exhaustiveness that no row carries "false" in both.

A	B	$A \vee B$	$\neg A \vee \neg B$	$\neg A \wedge \neg B$	$(\neg A \vee B) \wedge (\neg A \vee \neg B)$
true	true	true	false	false	false
true	false	true	true	false	false
false	true	true	true	false	true
false	false	false	true	true	true

Problem 2.6. Prove that $\alpha \models \beta$ iff $\alpha \wedge \neg\beta$ is inconsistent. This is known as the Refutation Theorem.

Solution. By Section 2.3.3, $\alpha \models \beta$ iff $\mathrm{Mods}(\alpha) \subseteq \mathrm{Mods}(\beta)$; by the identities $\mathrm{Mods}(\alpha \wedge \gamma) = \mathrm{Mods}(\alpha) \cap \mathrm{Mods}(\gamma)$ and $\mathrm{Mods}(\neg\beta) = \overline{\mathrm{Mods}(\beta)}$ of Section 2.3.1 (complement taken in the set Ω of all worlds),

$$\begin{aligned} \mathrm{Mods}(\alpha \wedge \neg\beta) &= \mathrm{Mods}(\alpha) \cap \overline{\mathrm{Mods}(\beta)} \\ &= \mathrm{Mods}(\alpha) \setminus \mathrm{Mods}(\beta). \end{aligned}$$

For any sets, $S \setminus T = \emptyset$ iff $S \subseteq T$. Hence the left-hand side is empty — that is, $\alpha \wedge \neg\beta$ is inconsistent in the sense of Section 2.3.2 — exactly when $\mathrm{Mods}(\alpha) \subseteq \mathrm{Mods}(\beta)$, i.e. exactly when $\alpha \models \beta$. ■

Problem 2.7. Prove that $\alpha \models \beta$ iff $\alpha \Rightarrow \beta$ is valid. This is known as the Deduction Theorem.

Solution. Since $\alpha \Rightarrow \beta$ abbreviates $\neg\alpha \vee \beta$ (Section 2.2), the model-set identities of Section 2.3.1 give

$$\begin{aligned} \mathrm{Mods}(\alpha \Rightarrow \beta) &= \mathrm{Mods}(\neg\alpha) \cup \mathrm{Mods}(\beta) \\ &= \overline{\mathrm{Mods}(\alpha)} \cup \mathrm{Mods}(\beta). \end{aligned}$$

For subsets $S, T \subseteq \Omega$ one has $\overline{S} \cup T = \Omega$ iff $S \subseteq T$, since the worlds outside $\overline{S}$ are precisely those in S. Taking $S = \mathrm{Mods}(\alpha)$ and $T = \mathrm{Mods}(\beta)$: validity of $\alpha \Rightarrow \beta$, i.e. $\mathrm{Mods}(\alpha \Rightarrow \beta) = \Omega$ (Section 2.3.2), holds iff $\mathrm{Mods}(\alpha) \subseteq \mathrm{Mods}(\beta)$, i.e. iff $\alpha \models \beta$. ■
Method (2): by Exercise 2.6, $\alpha \models \beta$ iff $\alpha \wedge \neg\beta$ is inconsistent, and a sentence γ is inconsistent iff $\neg\gamma$ is valid (complementation again). De Morgan and double negation (Table 2.2) turn $\neg(\alpha \wedge \neg\beta)$ into $\neg\alpha \vee \beta = \alpha \Rightarrow \beta$.

1.2 Exercises 2.8–2.13

Problem 2.8. Prove that if $\alpha \models \beta$, then $\alpha \wedge \beta$ is equivalent to α.

Solution. Assume $\alpha \models \beta$, that is, $\mathrm{Mods}(\alpha) \subseteq \mathrm{Mods}(\beta)$ (Section 2.3.3). Since $S \subseteq T$ implies $S \cap T = S$,

the identity of Section 2.3.1 gives

$$\begin{aligned}\mathrm{Mods}(\alpha \wedge \beta) &= \mathrm{Mods}(\alpha) \cap \mathrm{Mods}(\beta) \\ &= \mathrm{Mods}(\alpha),\end{aligned}$$

and sentences with identical model sets are equivalent. ■

Problem 2.9. Prove that if $\alpha \models \beta$, then $\alpha \vee \beta$ is equivalent to β.

Solution. Assume $\alpha \models \beta$, that is, $\mathrm{Mods}(\alpha) \subseteq \mathrm{Mods}(\beta)$ (Section 2.3.3). Since $S \subseteq T$ implies $S \cup T = T$, the identity of Section 2.3.1 gives

$$\begin{aligned}\mathrm{Mods}(\alpha \vee \beta) &= \mathrm{Mods}(\alpha) \cup \mathrm{Mods}(\beta) \\ &= \mathrm{Mods}(\beta),\end{aligned}$$

so $\alpha \vee \beta$ and β are equivalent. ■
Method (2): $\alpha \models \beta$ iff $\neg\beta \models \neg\alpha$, so Exercise 2.8 applied to that pair gives $\neg\beta \wedge \neg\alpha \equiv \neg\beta$; negating both sides and using de Morgan (Table 2.2) yields $\alpha \vee \beta \equiv \beta$.

Problem 2.10. Convert the following sentences into CNF:
(a) $P \Rightarrow (Q \Rightarrow R)$.
(b) $\neg((P \Rightarrow Q) \wedge (R \Rightarrow S))$.

Solution. (a) The single clause $\neg P \vee \neg Q \vee R$; (b) the four-clause CNF displayed below. Both parts run the procedure of Section 2.7.3: eliminate implications, push negations onto variables, distribute $\vee$ over $\wedge$.
(a) Eliminating the inner then the outer implication,

$$P \Rightarrow (Q \Rightarrow R) \;\equiv\; \neg P \vee (\neg Q \vee R) \;\equiv\; \neg P \vee \neg Q \vee R,$$

already in NNF, with no conjunction to distribute over.
(b) Eliminating implications, then de Morgan twice on the outer negation and double-negation removal,

$$\begin{aligned}\neg\big((P \Rightarrow Q) \wedge (R \Rightarrow S)\big) &\equiv \neg\big((\neg P \vee Q) \wedge (\neg R \vee S)\big) \\ &\equiv (P \wedge \neg Q) \vee (R \wedge \neg S).\end{aligned}$$

Distributing over each conjunction in turn,

$$\begin{aligned}(P \wedge \neg Q) \vee (R \wedge \neg S) \equiv{}& \big((P \wedge \neg Q) \vee R\big) \\ &\wedge \big((P \wedge \neg Q) \vee \neg S\big) \\ \equiv{}& (P \vee R) \wedge (\neg Q \vee R) \\ &\wedge (P \vee \neg S) \wedge (\neg Q \vee \neg S).\end{aligned}$$

No clause is a tautology or subsumed by another, so no clause can be dropped. The table compares $\varphi = \neg((P \Rightarrow Q) \wedge (R \Rightarrow S))$ with this CNF ψ over all sixteen worlds, writing $I_1 = P \Rightarrow Q$ and $I_2 = R \Rightarrow S$:

P	Q	R	S	I_1	I_2	φ	ψ
t	t	t	t	t	t	f	f
t	t	t	f	t	f	t	t
t	t	f	t	t	t	f	f
t	t	f	f	t	t	f	f
t	f	t	t	f	t	t	t
t	f	t	f	f	f	t	t
t	f	f	t	f	t	t	t
t	f	f	f	f	t	t	t
f	t	t	t	t	t	f	f
f	t	t	f	t	f	t	t
f	t	f	t	t	t	f	f
f	t	f	f	t	t	f	f
f	f	t	t	t	t	f	f
f	f	t	f	t	f	t	t
f	f	f	t	t	t	f	f
f	f	f	f	t	t	f	f

Problem 2.11. Let Γ be an NNF circuit that satisfies decomposability and determinism. Show how one can augment the circuit Γ with additional nodes so it also satisfies smoothness. What is the time and space complexity of your algorithm for ensuring smoothness?
Recall the definitions of Section 2.7. An NNF circuit is a rooted DAG whose leaves (inputs) are labeled with literals or with the constants *true* / *false*, and whose internal nodes are and-gates and or-gates. It is *decomposable* if for every and-node the sentences represented by any two of its children share no variables; *deterministic* if for every or-node the sentences represented by any two of its children are mutually exclusive; and *smooth* if for every or-node the sentences represented by any two of its children mention the same set of variables.

Solution. Conjoin onto each or-node edge the valid gadgets $X \vee \neg X$ for the variables its child is missing; with $m = |\Gamma|$ edges and v variables this costs $O(mv)$ time and yields a circuit of size $O(mv)$.
Write $\mathrm{Vars}(N)$ for the variables mentioned in the subcircuit at N, and assume every node is reachable from the root.

1. One reverse-topological sweep computes $\mathrm{Vars}(N)$ everywhere: $\emptyset$ at a constant input, $\{X\}$ at an input labeled X or $\neg X$, and $\bigcup_C \mathrm{Vars}(C)$ at an internal node. Store each set as a length-v bit vector.
2. Build once, per variable X, an or-node S_X with fresh children labeled X and $\neg X$; thus S_X is valid and $\mathrm{Vars}(S_X) = \{X\}$.
3. For each edge $N \to C$ with N an or-node, let $D = \mathrm{Vars}(N) \setminus \mathrm{Vars}(C)$; if $D \neq \emptyset$, redirect the edge to a fresh and-node
$$A_{N,C} \;=\; \mathrm{and}\Big(C,\ \{\, S_X : X \in D \,\}\Big).$$

The gadget must hang off the *edge*, not off C itself: in a DAG the same C may be a child of two or-nodes with different variable sets.
Equivalence. Each S_X is valid, so $A_{N,C}$ is equivalent to C by $\alpha \wedge \mathit{true} \equiv \alpha$ (Table 2.2), and replacing a child by an equivalent child preserves the sentence represented by every ancestor, hence by the root. Variable sets are unchanged, since $\mathrm{Vars}(C) \subseteq \mathrm{Vars}(N)$ (the latter being the union over the children) gives
$$\mathrm{Vars}(A_{N,C}) = \mathrm{Vars}(C) \cup D = \mathrm{Vars}(N);$$
so the sets from step 1 stay valid and the or-nodes may be processed independently, in one pass, in any order.
Smoothness holds because every child of an or-node N now mentions exactly $\mathrm{Vars}(N)$, and each S_X has both children mentioning $\{X\}$. Decomposability is preserved: the children of $A_{N,C}$ are C and the S_X with $X \in D$, pairwise variable-disjoint since $D \cap \mathrm{Vars}(C) = \emptyset$ and the gadgets are for

distinct variables, while original and-nodes keep their children and their variable sets. Determinism is preserved: equivalent replacements of mutually exclusive children remain mutually exclusive, and $X, \neg X$ are mutually exclusive.
Complexity. Step 1 performs one set union per edge, $O(mv)$ bit operations ($O(mv/w)$ machine words) and $O(mv)$ bits of working space; step 2 is $O(v)$; step 3 examines each edge once, computes a set difference in $O(v)$, and adds at most $1 + v$ edges. Hence $O(mv)$ time and $O(mv)$ output size, with $\sum_{N \to C} |\text{Vars}(N) \setminus \text{Vars}(C)|$ edges actually added — none on an already smooth circuit.

Problem 2.12. Let Γ be an NNF circuit that satisfies decomposability, is equivalent to CNF Δ, and does not contain *false*. Suppose that every model of Δ sets the same number of variables to true (we say in this case that all models of Δ have the same cardinality). Show that circuit Γ must be smooth.

Solution. Were some or-node N not smooth, a child C_1 would omit a variable $X \in \text{Vars}(N)$; extending a model of C_1 (Step 1) to $\text{Vars}(N)$ in the two ways ω_1, ω_2 that differ only at X gives two models of N, both satisfying N through C_1, with $\#\omega_1 = \#\omega_2 + 1$ — impossible by Step 4 below. Write $\text{Vars}(N)$ and $\text{Mods}(N)$ for the variables of the subcircuit at N and its models (assignments to $\text{Vars}(N)$), and $\#\sigma$ for the number of variables σ sets to true; every node is assumed reachable from the root.
Step 1: every node is satisfiable. Bottom-up: an input is *true* or a literal, never *false* by hypothesis, and each of these is satisfiable; an or-node is satisfied through any child; and at an and-node decomposability makes the children's variable sets pairwise disjoint with union $\text{Vars}(N)$, so models σ_i of the children have disjoint domains and $\sigma_1 \cup \cdots \cup \sigma_k$ is a well-defined model of N.
Step 2: local models extend coherently. For every node N and $\sigma, \sigma' \in \text{Mods}(N)$ there are models ω, ω' of Γ restricting to σ, σ' on $\text{Vars}(N)$ and agreeing with each other outside $\text{Vars}(N)$. Walk a path $N = N_0, \ldots, N_r$ to the root making the *same* choice on both branches at each step: at an or-parent extend both by one and the same assignment of $\text{Vars}(N_{j+1}) \setminus \text{Vars}(N_j)$, which satisfies N_{j+1} through the child N_j; at an and-parent adjoin one and the same model of each sibling (Step 1), legitimate because decomposability makes the domains disjoint.
Step 3: all models of Γ have one cardinality c. Γ is satisfiable by Step 1, so Δ has a model. If a variable X lay in exactly one of $V = \text{Vars}(\Delta)$ and $\text{Vars}(\Gamma)$, then over the common variable set $V \cup \text{Vars}(\Gamma)$ the sentence not mentioning X has a model set closed under flipping X; the two model sets are equal, so $\text{Mods}(\Delta)$ is closed under that flip too, and flipping changes cardinality by one, contradicting the hypothesis. Hence $\text{Vars}(\Gamma) = V$ and $\text{Mods}(\Gamma) = \text{Mods}(\Delta)$.
Step 4: every node inherits the property. For $\sigma, \sigma' \in \text{Mods}(N)$, take ω, ω' from Step 2; they agree outside $\text{Vars}(N)$, so

$$\#\sigma - \#\sigma' \;=\; \#\omega - \#\omega' \;=\; c - c \;=\; 0.$$

This contradicts $\#\omega_1 = \#\omega_2 + 1$ above, so no or-node child omits a variable of its parent: any two children of an or-node mention the same variables, i.e. Γ is smooth. ■

Problem 2.13. Let Γ be an NNF circuit that satisfies decomposability, determinism, and smoothness. Consider the following procedure for generating a subcircuit Γ_m of circuit Γ:

- Assign an integer to each node in circuit Γ as follows: an input node is assigned 0 if labeled with *true* or a positive literal, ∞ if labeled with *false*, and 1 if labeled with a negative literal. An or-node is assigned the minimum of integers assigned to its children, and an and-node is assigned the sum of integers assigned to its children.
- Obtain Γ_m from Γ by deleting every edge that extends from an or-node N to one of its children C, where N and C have different integers assigned to them.

Show that the models of Γ_m are the minimum-cardinality models of Γ, where the cardinality of a model is defined as the number of variables it sets to false.

Solution. The assigned integers are minimum cardinalities, and the deletion retains exactly the edges lying on a minimum-cardinality model. Write $\|\sigma\|$ for the number of variables σ sets to false, $d(N)$ for the integer at N, and $\text{Mods}(N)$ for the models of the subcircuit at N (assignments to $\text{Vars}(N)$);

arithmetic uses $\infty + x = \infty$, $\min(\infty, x) = x$, and $\min \emptyset = \infty$.

Lemma 1. $d(N) = \min\{\|\sigma\| : \sigma \in \mathrm{Mods}(N)\}$ at every node. Bottom-up. The inputs read off directly: 0 for *true* and for a positive literal, 1 for a negative literal, ∞ for *false* (empty model set). At an or-node, smoothness gives $\mathrm{Vars}(C_i) = \mathrm{Vars}(N)$ for all i, so $\mathrm{Mods}(N) = \bigcup_i \mathrm{Mods}(C_i)$ with no extension needed and

$$\min_{\sigma \in \mathrm{Mods}(N)} \|\sigma\| \;=\; \min_i d(C_i) \;=\; d(N);$$

smoothness is essential, since extending a child's model to $\mathrm{Vars}(N)$ could add false-valued variables and understate its cost. At an and-node, decomposability makes the $\mathrm{Vars}(C_i)$ pairwise disjoint with union $\mathrm{Vars}(N)$, so $\mathrm{Mods}(N)$ consists of the disjoint unions $\sigma_1 \cup \cdots \cup \sigma_k$, $\|\cdot\|$ is additive on them, and minimizing a sum of independent terms minimizes each term:

$$\min_{\sigma \in \mathrm{Mods}(N)} \|\sigma\| \;=\; \sum_i d(C_i) \;=\; d(N)$$

(both sides ∞ if some child is unsatisfiable).

Lemma 2. At an or-node the deleted children are exactly those with $d(C) > d(N)$, since $d(N) = \min_i d(C_i)$, and at least one child survives; by smoothness every child carries variable set $\mathrm{Vars}(N)$, so the surviving union is still $\mathrm{Vars}(N)$. And-node edges are never deleted, so $\mathrm{Vars}(N)$ is unchanged in Γ_m, which remains decomposable, deterministic, and smooth (a subset of an or-node's children preserves pairwise mutual exclusivity and pairwise equality of variable sets).

Main claim. For every node N,

$$\mathrm{Mods}_{\Gamma_m}(N) \;=\; \{\, \sigma \in \mathrm{Mods}_{\Gamma}(N) : \|\sigma\| = d(N) \,\},$$

both sides being sets of assignments to $\mathrm{Vars}(N)$ by Lemma 2. Bottom-up.

(i) *Inputs.* Untouched, and each has at most one model, of cardinality $d(N)$ by the base cases of Lemma 1; for *false* both sides are empty.

(ii) *Or-nodes.* By the inductive hypothesis and smoothness,

$$\begin{aligned}\mathrm{Mods}_{\Gamma_m}(N) &= \bigcup_{i\,:\,d(C_i)=d(N)} \{\sigma \in \mathrm{Mods}_{\Gamma}(C_i) : \|\sigma\| = d(N)\} \\ &= \bigcup_{i=1}^{k} \{\sigma \in \mathrm{Mods}_{\Gamma}(C_i) : \|\sigma\| = d(N)\},\end{aligned}$$

because a deleted child has $d(C_i) > d(N)$ and by Lemma 1 all of its models cost at least $d(C_i)$, so it contributes nothing; the last union is $\{\sigma \in \mathrm{Mods}_{\Gamma}(N) : \|\sigma\| = d(N)\}$.

(iii) *And-nodes.* All children survive, models of N are the disjoint unions $\bigcup_i \sigma_i$ with $\|\bigcup_i \sigma_i\| = \sum_i \|\sigma_i\|$, and $d(N) = \sum_i d(C_i)$ with $\|\sigma_i\| \geq d(C_i)$ termwise (Lemma 1); so the sums agree only if $\|\sigma_i\| = d(C_i)$ for every i, which is exactly what the inductive hypothesis says $\mathrm{Mods}_{\Gamma_m}(C_i)$ contains. (Both sides empty if some $d(C_i) = \infty$.)

At the root, Lemma 1 identifies $d(\text{root})$ with $\min_{\tau \in \mathrm{Mods}(\Gamma)} \|\tau\|$, so the models of Γ_m are precisely the minimum-cardinality models of Γ. ■

2 Probability Calculus

Contents

2.1 Exercises 3.1–3.7 12
2.2 Exercises 3.8–3.14 15
2.3 Exercises 3.15–3.21 20
2.4 Exercises 3.22–3.27 23

2.1 Exercises 3.1–3.7

Problem 3.1. Consider the following joint distribution over the propositional variables A, B, C.

world	A	B	C	Pr(.)
ω_1	true	true	true	.075
ω_2	true	true	false	.050
ω_3	true	false	true	.225
ω_4	true	false	false	.150
ω_5	false	true	true	.025
ω_6	false	true	false	.100
ω_7	false	false	true	.075
ω_8	false	false	false	.300

(a) What is $\Pr(A = \text{true})$? $\Pr(B = \text{true})$? $\Pr(C = \text{true})$?
(b) Update the distribution by conditioning on the event $C = \text{true}$, that is, construct the conditional distribution $\Pr(. \mid C = \text{true})$.
(c) What is $\Pr(A = \text{true} \mid C = \text{true})$? $\Pr(B = \text{true} \mid C = \text{true})$?
(d) Is the event $A = \text{true}$ independent of the event $C = \text{true}$? Is $B = \text{true}$ independent of $C = \text{true}$?

Solution. (a) .5, .25, .4; (b) the table below; (c) .75 and .25; (d) $A = \text{true}$ is not independent of $C = \text{true}$, while $B = \text{true}$ is.
(a) Equation (3.1) sums the beliefs of the worlds satisfying each event:

$$\Pr(A = \text{true}) = .075 + .050 + .225 + .150 = .5,$$
$$\Pr(B = \text{true}) = .075 + .050 + .025 + .100 = .25,$$
$$\Pr(C = \text{true}) = .075 + .225 + .025 + .075 = .4.$$

(b) By (3.11) every world falsifying the evidence gets belief 0 and every world satisfying it is rescaled by $\Pr(C = \text{true}) = .4$, e.g. $.075/.4 = .1875$:

world	A	B	C	$\Pr(. \mid C = \text{true})$
ω_1	true	true	true	.1875
ω_2	true	true	false	0
ω_3	true	false	true	.5625
ω_4	true	false	false	0
ω_5	false	true	true	.0625
ω_6	false	true	false	0
ω_7	false	false	true	.1875
ω_8	false	false	false	0

(c) By (3.12), equivalently by summing the surviving entries of (b),

$$\Pr(A = \text{true} \mid C = \text{true}) = \frac{.075 + .225}{.4} = .75,$$

$$\Pr(B = \text{true} \mid C = \text{true}) = \frac{.075 + .025}{.4} = .25.$$

(d) By Definition (3.13), α is independent of β when $\Pr(\alpha \mid \beta) = \Pr(\alpha)$. Here $.75 \neq .5 = \Pr(A = \text{true})$, so $A = \text{true}$ and $C = \text{true}$ are not independent, while $.25 = \Pr(B = \text{true})$, so $B = \text{true}$ and $C = \text{true}$ are. The symmetric form (3.14) agrees: $\Pr(A = \text{true} \wedge C = \text{true}) = .3 \neq .2 = (.5)(.4)$, whereas $\Pr(B = \text{true} \wedge C = \text{true}) = .1 = (.25)(.4)$.

Problem 3.2. Consider again the joint distribution Pr from Exercise 3.1, repeated here for convenience.

world	A	B	C	Pr(.)
ω_1	true	true	true	.075
ω_2	true	true	false	.050
ω_3	true	false	true	.225
ω_4	true	false	false	.150
ω_5	false	true	true	.025
ω_6	false	true	false	.100
ω_7	false	false	true	.075
ω_8	false	false	false	.300

(a) What is $\Pr(A = \text{true} \vee B = \text{true})$?
(b) Update the distribution by conditioning on the event $A = \text{true} \vee B = \text{true}$, that is, construct the conditional distribution $\Pr(. \mid A = \text{true} \vee B = \text{true})$.
(c) What is $\Pr(A = \text{true} \mid A = \text{true} \vee B = \text{true})$? $\Pr(B = \text{true} \mid A = \text{true} \vee B = \text{true})$?
(d) Determine whether the event $B = \text{true}$ is conditionally independent of $C = \text{true}$ given the event $A = \text{true} \vee B = \text{true}$.

Solution. (a) .625; (b) the table below; (c) .8 and .4; (d) no. Write β for the evidence $A = \text{true} \vee B = \text{true}$.
(a) The worlds satisfying β are $\omega_1, \ldots, \omega_6$, so by (3.1)

$$\Pr(\beta) = .075 + .050 + .225 + .150 + .025 + .100 = .625.$$

(b) By (3.11) the two falsifying worlds ω_7, ω_8 get belief 0 and the rest are divided by .625, e.g. $.075/.625 = .12$:

world	A	B	C	$\Pr(. \mid \beta)$
ω_1	true	true	true	.12
ω_2	true	true	false	.08
ω_3	true	false	true	.36
ω_4	true	false	false	.24
ω_5	false	true	true	.04
ω_6	false	true	false	.16
ω_7	false	false	true	0
ω_8	false	false	false	0

(c) A world with A true already satisfies β, so $(A = \text{true}) \wedge \beta \equiv (A = \text{true})$, and likewise for B; hence by (3.12)

$$\Pr(A = \text{true} \mid \beta) = \frac{.5}{.625} = .8, \qquad \Pr(B = \text{true} \mid \beta) = \frac{.25}{.625} = .4.$$

(d) Not conditionally independent. By (3.16) the test is whether $\Pr(B = \text{true} \wedge C = \text{true} \mid \beta)$ equals the product of the two separate conditionals. The worlds with B and C true are ω_1, ω_5, both satisfying β; those with C true satisfying β are $\omega_1, \omega_3, \omega_5$. So

$$\Pr(B = \text{true} \wedge C = \text{true} \mid \beta) = \frac{.1}{.625} = .16,$$

$$\Pr(C = \text{true} \mid \beta) = \frac{.325}{.625} = .52,$$

while $\Pr(B = \text{true} \mid \beta) \Pr(C = \text{true} \mid \beta) = (.4)(.52) = .208 \neq .16$.

Problem 3.3. Suppose that we tossed two unbiased coins C_1 and C_2.
(a) Given that the first coin landed heads, $C_1 = h$, what is the probability that the second coin landed tails, $\Pr(C_2 = t \mid C_1 = h)$?
(b) Given that at least one of the coins landed heads, $C_1 = h \vee C_2 = h$, what is the probability that both coins landed heads, $\Pr(C_1 = h \wedge C_2 = h \mid C_1 = h \vee C_2 = h)$?

Solution. Unbiasedness gives $\Pr(C_1 = h) = \Pr(C_2 = h) = 1/2$, and tossing two separate coins makes the outcomes independent, so by (3.14) each world carries the product $1/4$:

world	C_1	C_2	$\Pr(.)$
ω_1	h	h	$1/4$
ω_2	h	t	$1/4$
ω_3	t	h	$1/4$
ω_4	t	t	$1/4$

(a) $1/2$. The evidence holds in ω_1, ω_2 and $C_2 = t \wedge C_1 = h$ only in ω_2, so by Bayes conditioning (3.12),

$$\Pr(C_2 = t \mid C_1 = h) = \frac{1/4}{1/2} = \frac{1}{2}.$$

(b) $1/3$. Put $\beta : C_1 = h \vee C_2 = h$, holding in $\omega_1, \omega_2, \omega_3$, and $\alpha : C_1 = h \wedge C_2 = h$, holding only in ω_1; since $\alpha \models \beta$ we have $\alpha \wedge \beta \equiv \alpha$, so by (3.12)

$$\Pr(\alpha \mid \beta) = \frac{1/4}{3/4} = \frac{1}{3}.$$

Problem 3.4. Suppose that 24% of a population are smokers and that 5% of the population have cancer. Suppose further that 86% of the population with cancer are also smokers. What is the probability that a smoker will also have cancer?

Solution. About 17.9%. Writing S for "the individual smokes" and C for "the individual has cancer" under uniform sampling from the population, the data read $\Pr(S) = .24$, $\Pr(C) = .05$, $\Pr(S \mid C) = .86$, and Bayes rule (3.19) supplies the reverse conditional:

$$\Pr(C \mid S) = \frac{\Pr(S \mid C)\,\Pr(C)}{\Pr(S)} = \frac{(.86)(.05)}{.24} = \frac{.043}{.24} \approx .1792.$$

Problem 3.5. Consider again the population from Exercise 3.4, in which 24% are smokers, 5% have cancer, and 86% of those with cancer are smokers. What is the relative change in the odds that a member of the population has cancer upon learning that they are also a smoker?

Solution. The odds of cancer are multiplied by $k \approx 4.147$, rising from about 1:19 to about 1:4.58. Keeping S and C and the data of Exercise 3.4, the quantity asked for is the Bayes factor $k = O'(C)/O(C)$, with $O(\gamma) = \Pr(\gamma)/\Pr(\neg\gamma)$ the odds (3.23) and O' computed in $\Pr' = \Pr(. \mid S)$. From Exercise 3.4, $\Pr(C \mid S) = .043/.24$, so $\Pr(\neg C \mid S) = (.24 - .043)/.24 = .197/.24$ and the common denominator $\Pr(S)$ cancels:

$$k = \frac{O'(C)}{O(C)} = \frac{.043/.197}{.05/.95} = \frac{(.043)(.95)}{(.197)(.05)} = \frac{817}{197} \approx 4.147.$$

Method (2): applying (3.19) to numerator and denominator of $O'(C)$ cancels $\Pr(S)$ and gives $k = \Pr(S \mid C)/\Pr(S \mid \neg C)$; case analysis (3.17) yields $\Pr(S \wedge \neg C) = .24 - .043 = .197$, so $\Pr(S \mid \neg C) = .197/.95$ and $k = (.86)(.95)/.197 \approx 4.147$.

Problem 3.6. Consider a family with two children, ages four and nine.
(a) What is the probability that the older child is a boy?
(b) What is the probability that the older child is a boy given that the younger child is a boy?
(c) What is the probability that the older child is a boy given that at least one of the children is a boy?
(d) What is the probability that both children are boys given that at least one of them is a boy?
Define your variables and the corresponding joint probability distribution. Moreover, for each of these questions define α and β for which $\Pr(\alpha \mid \beta)$ is the answer.

Solution. Take O for the sex of the older child and Y for that of the younger, each with values b and g; the differing ages make the children distinguishable. Under the usual idealization each child is equally likely a boy or a girl and the two sexes are independent, so by (3.14) every world carries the product $1/2 \times 1/2$:

world	O	Y	Pr(.)
ω_1	b	b	$1/4$
ω_2	b	g	$1/4$
ω_3	g	b	$1/4$
ω_4	g	g	$1/4$

The event "at least one child is a boy" is $O = b \vee Y = b$, holding in $\omega_1, \omega_2, \omega_3$, of belief $3/4$.
(a) $\alpha : O = b$, $\beta : \top$; since $\Pr(\top) = 1$, the answer is $\Pr(O = b) = 1/4 + 1/4 = 1/2$.
(b) $\alpha : O = b$, $\beta : Y = b$. The evidence holds in ω_1, ω_3 and the conjunction only in ω_1, so by (3.12) the answer is $(1/4)/(1/2) = 1/2$ — the independence assumption showing itself.
(c) $\alpha : O = b$, $\beta : O = b \vee Y = b$. Here $\alpha \models \beta$, so $\alpha \wedge \beta \equiv \alpha$ and

$$\Pr(O = b \mid O = b \vee Y = b) = \frac{1/2}{3/4} = \frac{2}{3}.$$

(d) $\alpha : O = b \wedge Y = b$, $\beta : O = b \vee Y = b$. Again $\alpha \models \beta$, so

$$\Pr(O = b \wedge Y = b \mid O = b \vee Y = b) = \frac{1/4}{3/4} = \frac{1}{3}.$$

Problem 3.7. Prove Equation 3.19, that is, Bayes rule:

$$\Pr(\alpha \mid \beta) = \frac{\Pr(\beta \mid \alpha)\ \Pr(\alpha)}{\Pr(\beta)}.$$

Solution. Bayes conditioning (3.12) applied twice to the same joint belief. Assume $\Pr(\alpha) \neq 0$ and $\Pr(\beta) \neq 0$, without which $\Pr(\beta \mid \alpha)$ and the right-hand side are undefined. Applying (3.12) with evidence α and multiplying through,

$$\Pr(\beta \mid \alpha)\ \Pr(\alpha) = \Pr(\beta \wedge \alpha) \ = \ \Pr(\alpha \wedge \beta),$$

the second equality because $\alpha \wedge \beta$ and $\beta \wedge \alpha$ have the same models and (3.1) reads belief off the model set alone. Substituting that numerator into (3.12) with evidence β,

$$\Pr(\alpha \mid \beta) = \frac{\Pr(\alpha \wedge \beta)}{\Pr(\beta)} = \frac{\Pr(\beta \mid \alpha)\ \Pr(\alpha)}{\Pr(\beta)}. \qquad \blacksquare$$

2.2 Exercises 3.8–3.14

Problem 3.8. Suppose that we have a patient who was just tested for a particular disease and the test came out positive. We know that one in every thousand people has this disease. We also know that the test is not reliable: it has a false positive rate of 2% and a false negative rate of 5%. We have seen previously that the probability of having the disease is $\approx 4.5\%$ given a positive test result. Suppose that the test is repeated n times and all tests come out positive. What is the smallest n for which the belief in the disease is greater than 95%, assuming the errors of various tests are independent? Justify your answer.

Solution. $n = 3$. Let D be the disease and T_i the event that test i reads positive, so $\Pr(D) = .001$, $f_p = \Pr(T_i \mid \neg D) = .02$ and $f_n = \Pr(\neg T_i \mid D) = .05$. "Errors independent" means the readings are mutually independent given D and given $\neg D$ — conditional independence (3.16), not unconditional

independence — so for $e_n = T_1 \wedge \cdots \wedge T_n$,

$$\Pr(e_n \mid D) = (.95)^n, \qquad \Pr(e_n \mid \neg D) = (.02)^n.$$

Applying (3.12) to numerator and denominator of the odds, each reading contributes the noisy-sensor Bayes factor $k^+ = (1 - f_n)/f_p = 47.5$ of Section 3.6.4:

$$O(D \mid e_n) = \frac{\Pr(e_n \mid D)\Pr(D)}{\Pr(e_n \mid \neg D)\Pr(\neg D)} = (47.5)^n \cdot \frac{.001}{.999} = \frac{(47.5)^n}{999}.$$

A belief exceeds 95% exactly when its odds exceed $.95/.05 = 19$, so we need $(47.5)^n > 18981$, i.e. $n > \ln 18981 / \ln 47.5 \approx 2.552$; since $(47.5)^n$ increases with n, the least such integer is $n = 3$. Converting back with $\Pr = O/(1+O)$:

n	$O(D \mid e_n) = 47.5^n/999$	$\Pr(D \mid e_n)$
1	.047548	.04539
2	2.25851	.69311
3	107.279	.99076

Problem 3.9. Consider the following distribution over three variables:

world	A	B	C	Pr(.)
ω_1	true	true	true	.27
ω_2	true	true	false	.18
ω_3	true	false	true	.03
ω_4	true	false	false	.02
ω_5	false	true	true	.02
ω_6	false	true	false	.03
ω_7	false	false	true	.18
ω_8	false	false	false	.27

For each pair of variables, state whether they are independent. State also whether they are independent given the third variable. Justify your answers.

Solution. No pair is marginally independent, and the only conditional independence of the three is $B \perp C \mid A$. Write a for $A =$ true, $\bar{a}$ for $A =$ false, similarly for B and C; by (3.14) a pair of variables is independent only if $\Pr(x, y) = \Pr(x)\Pr(y)$ at *every* instantiation, so one violation refutes it.
Summing the worlds by (3.1) gives $\Pr(a) = \Pr(b) = \Pr(c) = .50$, so independence of a pair would force each joint entry to be .25. It does not:

$$\Pr(a, b) = .27 + .18 = .45,$$
$$\Pr(a, c) = .27 + .03 = .30,$$
$$\Pr(b, c) = .27 + .02 = .29.$$

(i) B and C are independent given A. Conditioning on a (of probability .50), with $\Pr(b \mid a) = (.27 + .18)/.5 = .90$ and $\Pr(c \mid a) = (.27 + .03)/.5 = .60$:

BC	$\Pr(\cdot \mid a)$	product of the two marginals
b, c	.54	$(.90)(.60) = .54$
$b, \bar{c}$	.36	$(.90)(.40) = .36$
$\bar{b}, c$	.06	$(.10)(.60) = .06$
$\bar{b}, \bar{c}$	.04	$(.10)(.40) = .04$

Conditioning on $\bar{a}$ (also .50), with $\Pr(b \mid \bar{a}) = (.02 + .03)/.5 = .10$ and $\Pr(c \mid \bar{a}) = (.02 + .18)/.5 = .40$:

BC	$\Pr(\cdot \mid \bar{a})$	product of the two marginals
b, c	.04	$(.10)(.40) = .04$
$b, \bar{c}$	.06	$(.10)(.60) = .06$
$\bar{b}, c$	.36	$(.90)(.40) = .36$
$\bar{b}, \bar{c}$	.54	$(.90)(.60) = .54$

Every entry of both tables factorizes, so by (3.16) $I_{\Pr}(B, A, C)$ holds.

(ii) A and B are *not* independent given C. Conditioning on c, with $\Pr(a \mid c) = (.27 + .03)/.5 = .60$ and $\Pr(b \mid c) = (.27 + .02)/.5 = .58$,

$$\Pr(a, b \mid c) = \frac{.27}{.50} = .54 \qquad \text{but} \qquad (.60)(.58) = .348.$$

(iii) A and C are *not* independent given B. Conditioning on b, with $\Pr(a \mid b) = (.27 + .18)/.5 = .90$ and $\Pr(c \mid b) = (.27 + .02)/.5 = .58$,

$$\Pr(a, c \mid b) = \frac{.27}{.50} = .54 \qquad \text{but} \qquad (.90)(.58) = .522.$$

Problem 3.10. Show the following:
(a) If $\alpha \models \beta$ and $\Pr(\beta) = 0$, then $\Pr(\alpha) = 0$.
(b) $\Pr(\alpha \wedge \beta) \leq \Pr(\alpha) \leq \Pr(\alpha \vee \beta)$.
(c) If $\alpha \models \beta$, then $\Pr(\alpha) \leq \Pr(\beta)$.
(d) If $\alpha \models \beta \models \gamma$, then $\Pr(\alpha \mid \beta) \geq \Pr(\alpha \mid \gamma)$.

Solution. All four parts follow from monotonicity: if $\alpha \models \beta$ then $\Pr(\alpha) \leq \Pr(\beta)$. Indeed $\alpha \models \beta$ says $\text{Mods}(\alpha) \subseteq \text{Mods}(\beta)$, so splitting the sum (3.1) along that inclusion,

$$\Pr(\beta) \;=\; \Pr(\alpha) + \sum_{\omega \models \beta \wedge \neg\alpha} \Pr(\omega) \;\geq\; \Pr(\alpha),$$

the second term being a sum of nonnegative beliefs. That is (c), with the sharper identity $\Pr(\beta) - \Pr(\alpha) = \Pr(\beta \wedge \neg\alpha)$.
(a) By (c), $\Pr(\alpha) \leq \Pr(\beta) = 0$, while $\Pr(\alpha) \geq 0$ by (3.2).
(b) Two applications of (c), to $\alpha \wedge \beta \models \alpha$ and to $\alpha \models \alpha \vee \beta$.
(d) Assume $\Pr(\beta) > 0$, as both conditionals require; $\beta \models \gamma$ and (c) then give $\Pr(\gamma) \geq \Pr(\beta) > 0$, so $\Pr(\alpha \mid \gamma)$ is defined too. From $\alpha \models \beta$, $\text{Mods}(\alpha \wedge \beta) = \text{Mods}(\alpha) \cap \text{Mods}(\beta) = \text{Mods}(\alpha)$, i.e. $\alpha \wedge \beta \equiv \alpha$; transitivity of $\models$ gives $\alpha \models \gamma$ and hence $\alpha \wedge \gamma \equiv \alpha$ as well. So by (3.12),

$$\Pr(\alpha \mid \beta) = \frac{\Pr(\alpha)}{\Pr(\beta)} \;\geq\; \frac{\Pr(\alpha)}{\Pr(\gamma)} = \Pr(\alpha \mid \gamma),$$

since $0 < \Pr(\beta) \leq \Pr(\gamma)$ and $\Pr(\alpha) \geq 0$.

Problem 3.11. Let α and β be two propositional sentences over disjoint sets of variables $\mathbf{X}$ and $\mathbf{Y}$, respectively. Show that α and β are independent, that is,

$$\Pr(\alpha \wedge \beta) = \Pr(\alpha)\Pr(\beta),$$

if the variable sets $\mathbf{X}$ and $\mathbf{Y}$ are independent, that is, $\Pr(\mathbf{x}, \mathbf{y}) = \Pr(\mathbf{x})\Pr(\mathbf{y})$ for all instantiations $\mathbf{x}$ of $\mathbf{X}$ and $\mathbf{y}$ of $\mathbf{Y}$.

Solution. Group the worlds by their $\mathbf{X}$- and $\mathbf{Y}$-parts, then apply the hypothesis. As $\mathbf{X} \cap \mathbf{Y} = \emptyset$, every world splits uniquely as $\omega = (\mathbf{x}, \mathbf{y}, \mathbf{z})$ with $\mathbf{z}$ the instantiation of the remaining variables; and since α mentions only $\mathbf{X}$, its truth value at ω depends on $\mathbf{x}$ alone, so $\omega \models \alpha$ iff $\mathbf{x} \models \alpha$, and likewise $\omega \models \beta$ iff $\mathbf{y} \models \beta$. Hence by (3.1), regrouping finite sums of nonnegative terms so that the omitted variables are summed out into marginals,

$$\Pr(\alpha) = \sum_{\mathbf{x} \models \alpha} \Pr(\mathbf{x}), \qquad \Pr(\beta) = \sum_{\mathbf{y} \models \beta} \Pr(\mathbf{y}).$$

The same regrouping, followed by the hypothesis $\Pr(\mathbf{x},\mathbf{y}) = \Pr(\mathbf{x})\Pr(\mathbf{y})$, factors the double sum:

$$\begin{aligned}\Pr(\alpha \wedge \beta) &= \sum_{\mathbf{x}\models\alpha}\sum_{\mathbf{y}\models\beta} \Pr(\mathbf{x})\Pr(\mathbf{y}) \\ &= \Big(\sum_{\mathbf{x}\models\alpha} \Pr(\mathbf{x})\Big)\Big(\sum_{\mathbf{y}\models\beta} \Pr(\mathbf{y})\Big) \;=\; \Pr(\alpha)\;\Pr(\beta),\end{aligned}$$

which is the definition (3.14) of independence of the events. ■
The identical regrouping over k pairwise disjoint blocks $\mathbf{X}_1,\ldots,\mathbf{X}_k$ whose joint distribution factorizes gives $\Pr(\alpha_1 \wedge \cdots \wedge \alpha_k) = \prod_i \Pr(\alpha_i)$ for sentences α_i over $\mathbf{X}_i$; Exercise 3.12 needs that form at its and-nodes.

Problem 3.12. Consider a propositional sentence α that is represented by an NNF circuit that satisfies the properties of decomposability and determinism. (Recall from Section 2.7 that an NNF circuit is a DAG whose leaves are labeled with literals or with the constants true/false and whose internal nodes are and-gates and or-gates; it is *decomposable* if for every and-node and every pair of its children C_1, C_2 the sentences represented by C_1 and C_2 share no variables, and *deterministic* if for every or-node and every pair of its children C_1, C_2 the sentences represented by C_1 and C_2 are mutually exclusive.) Suppose the circuit inputs are over variables $X_1,\ldots,X_n$ and that each variable X_i is independent of every other set of variables that does not contain X_i. Show that if given the probability distribution $\Pr(x_i)$ for each variable X_i, the probability of α can be computed in time linear in the size of the NNF circuit.

Solution. Replace each literal input by its probability, each and-gate by multiplication and each or-gate by addition, then evaluate the resulting arithmetic circuit bottom-up with the values cached at the nodes; one pass costs $O(|\Gamma|)$.
The distribution factorizes. Each X_{k+1} is independent of $\{X_1,\ldots,X_k\}$, so the event form (3.14) gives $\Pr(x_1,\ldots,x_{k+1}) = \Pr(x_{k+1})\Pr(x_1,\ldots,x_k)$, and induction on k yields

$$\Pr(x_1,\ldots,x_n) = \prod_{i=1}^{n} \Pr(x_i)$$

for every instantiation: the n given marginals specify Pr completely. Summing this over the variables outside pairwise disjoint blocks $\mathbf{Y}_1,\ldots,\mathbf{Y}_k$ shows the marginal factorizes too, so the k-block form of Exercise 3.11 applies: for sentences α_j over $\mathbf{Y}_j$,

$$\Pr(\alpha_1 \wedge \cdots \wedge \alpha_k) = \prod_{j=1}^{k} \Pr(\alpha_j). \qquad (\dagger)$$

The algorithm. In topological order (children before parents) store $v(N) = \Pr(\ell)$ at a leaf labeled with literal ℓ, $v(N) = 1$ at *true* and 0 at *false*, $v(N) = \prod_j v(C_j)$ at an and-node, and $v(N) = \sum_j v(C_j)$ at an or-node; output $v(\text{root})$.
Correctness. $v(N) = \Pr(\alpha_N)$, by induction along that order. Leaves are immediate, the constants by (3.4) and (3.3). At an and-node, decomposability makes the variable sets $\mathbf{Y}_j$ of the children pairwise disjoint, so (†) applies and $\Pr(\alpha_N) = \prod_j \Pr(\alpha_{C_j}) = \prod_j v(C_j)$. At an or-node, determinism makes the $\text{Mods}(\alpha_{C_j})$ pairwise disjoint with union $\text{Mods}(\alpha_N)$, so the sum (3.1) splits along that disjoint union:

$$\Pr(\alpha_N) = \sum_{j=1}^{k} \sum_{\omega\models\alpha_{C_j}} \Pr(\omega) = \sum_{j=1}^{k} \Pr(\alpha_{C_j}) = \sum_{j=1}^{k} v(C_j) = v(N).$$

At the root this gives $\Pr(\alpha)$.
Complexity. Caching means each node is visited once and performs one arithmetic operation per child edge, for a total of $O(|\text{nodes}(\Gamma)| + |\text{edges}(\Gamma)|) = O(|\Gamma|)$; the topological order costs the same and each lookup $\Pr(x_i)$ is $O(1)$. (Without caching, a shared subcircuit would be re-evaluated once per parent, costing the size of the unfolded tree.)

Problem 3.13. (After Pearl) We have three urns labeled 1, 2, and 3. The urns contain, respectively, three white and three black balls, four white and two black balls, and one white and two black balls. An experiment consists of selecting an urn at random then drawing a ball from it.

(a) Define the set of worlds that correspond to the various outcomes of this experiment. Assume you have two variables U with values 1, 2, and 3 and C with values black and white.

(b) Define the joint probability distribution over the set of possible worlds identified in (a).

(c) Find the probability of drawing a black ball.

(d) Find the conditional probability that urn 2 was selected given that a black ball was drawn.

(e) Find the probability of selecting urn 1 or a white ball.

Solution. (a) The worlds are the $3 \times 2 = 6$ instantiations of $U \in \{1, 2, 3\}$ and $C \in \{\text{b}, \text{w}\}$ (black, white):

$$\begin{aligned} \omega_1 &= (U{=}1, C{=}\text{b}), & \omega_2 &= (U{=}1, C{=}\text{w}), \\ \omega_3 &= (U{=}2, C{=}\text{b}), & \omega_4 &= (U{=}2, C{=}\text{w}), \\ \omega_5 &= (U{=}3, C{=}\text{b}), & \omega_6 &= (U{=}3, C{=}\text{w}). \end{aligned}$$

(b) "At random" means $\Pr(U{=}1) = \Pr(U{=}2) = \Pr(U{=}3) = 1/3$, and given the urn the ball is drawn uniformly from its contents:

urn	white	black	total	$\Pr(\text{b} \mid U)$	$\Pr(\text{w} \mid U)$
1	3	3	6	$3/6 = 1/2$	$1/2$
2	4	2	6	$2/6 = 1/3$	$2/3$
3	1	2	3	$2/3$	$1/3$

By the chain rule, $\Pr(u, c) = \Pr(c \mid u) \Pr(u)$, giving the joint distribution:

world	U	C	$\Pr(.)$
ω_1	1	black	$\frac{1}{3} \cdot \frac{1}{2} = \frac{1}{6}$
ω_2	1	white	$\frac{1}{3} \cdot \frac{1}{2} = \frac{1}{6}$
ω_3	2	black	$\frac{1}{3} \cdot \frac{1}{3} = \frac{1}{9}$
ω_4	2	white	$\frac{1}{3} \cdot \frac{2}{3} = \frac{2}{9}$
ω_5	3	black	$\frac{1}{3} \cdot \frac{2}{3} = \frac{2}{9}$
ω_6	3	white	$\frac{1}{3} \cdot \frac{1}{3} = \frac{1}{9}$

(The six entries sum to $18/18 = 1$. Check!)

(c) Summing the black worlds by (3.1),

$$\Pr(C{=}\text{b}) = \tfrac{1}{6} + \tfrac{1}{9} + \tfrac{2}{9} = \tfrac{3+2+4}{18} = \tfrac{1}{2}.$$

(d) By Bayes conditioning (3.12),

$$\Pr(U{=}2 \mid C{=}\text{b}) = \frac{\Pr(U{=}2, C{=}\text{b})}{\Pr(C{=}\text{b})} = \frac{1/9}{1/2} = \frac{2}{9} \approx .2222.$$

(e) For $\alpha = (U{=}1) \vee (C{=}\text{w})$, inclusion-exclusion (3.6) with $\Pr(C{=}\text{w}) = 1 - \Pr(C{=}\text{b}) = 1/2$ by (3.5) gives

$$\Pr(\alpha) = \tfrac{1}{3} + \tfrac{1}{2} - \tfrac{1}{6} = \tfrac{2+3-1}{6} = \tfrac{2}{3}.$$

Problem 3.14. Suppose we are presented with two urns labeled 1 and 2 and we want to distribute k white balls and k black balls between these urns. In particular, say that we want to pick an n and m where we place n white balls and m black balls into urn 1 and the remaining $k-n$ white balls and $k-m$ black balls into urn 2. Once we distribute the balls to urns, say that we play a game where we pick an urn at random and draw a ball from it.

(a) What is the probability that we draw a white ball for a given n and m?

Suppose now that we want to choose n and m so that we maximize the probability that we draw a white ball. Clearly, if both urns have an equal number of white and black balls (i.e., $n = m$), then the probability that we draw a white ball is $\frac{1}{2}$.

(b) Suppose that $k = 3$. Can we choose an n and m so that we increase the probability of drawing a white ball to $\frac{7}{10}$?

(c) Can we design a strategy for choosing n and m so that as k tends to infinity, the probability of drawing a white ball tends to $\frac{3}{4}$?

Solution. (a) Writing U for the urn picked and C for the color drawn, case analysis (3.18) over U gives

$$\Pr(C{=}\mathrm{w}) = \frac{1}{2} \cdot \frac{n}{n+m} + \frac{1}{2} \cdot \frac{k-n}{2k-n-m},$$

valid whenever both urns are nonempty, i.e. $(n, m) \neq (0, 0)$ and $(n, m) \neq (k, k)$.
(b) Yes: take $n = 1$, $m = 0$, a single white ball alone in urn 1 and the rest (2 white, 3 black) in urn 2. Then

$$\Pr(C{=}\mathrm{w}) = \frac{1}{2} \cdot \frac{1}{1} + \frac{1}{2} \cdot \frac{2}{5} = \frac{1}{2} + \frac{1}{5} = \frac{7}{10}.$$

(c) Yes: use the same rule $n = 1$, $m = 0$ for every k, so that urn 2 holds $k-1$ white and k black balls. Then

$$p_k := \Pr(C{=}\mathrm{w}) = \frac{1}{2} + \frac{k-1}{2(2k-1)} = \frac{3k-2}{4k-2},$$

and dividing numerator and denominator by k,

$$\lim_{k\to\infty} p_k = \lim_{k\to\infty} \frac{3 - 2/k}{4 - 2/k} = \frac{3}{4}.$$

(Consistently $p_3 = 7/10$, the value found in (b).)

2.3 Exercises 3.15–3.21

Problem 3.15. Prove the equivalence between the two definitions of conditional independence given by Equations 3.15 and 3.16. That is, show that for any events α, β, γ and any state of belief Pr, the condition

$$Pr(\alpha \mid \beta \wedge \gamma) = Pr(\alpha \mid \gamma) \quad \text{or} \quad Pr(\beta \wedge \gamma) = 0 \tag{3.15}$$

holds if and only if

$$Pr(\alpha \wedge \beta \mid \gamma) = Pr(\alpha \mid \gamma) Pr(\beta \mid \gamma) \quad \text{or} \quad Pr(\gamma) = 0. \tag{3.16}$$

Solution. Three exhaustive cases, in each of which (3.15) and (3.16) hold or fail together.
(i) $Pr(\gamma) = 0$. Then (3.16) holds by its second disjunct; and $\beta \wedge \gamma \models \gamma$, so monotonicity (Exercise 3.10(c)) gives $Pr(\beta \wedge \gamma) \leq Pr(\gamma) = 0$ and (3.15) holds by its second disjunct.
(ii) $Pr(\gamma) > 0$ but $Pr(\beta \wedge \gamma) = 0$. Then (3.15) holds by its second disjunct. For (3.16), $\alpha \wedge \beta \wedge \gamma \models \beta \wedge \gamma$ forces $Pr(\alpha \wedge \beta \wedge \gamma) = 0$, so

$$Pr(\alpha \wedge \beta \mid \gamma) = 0 = Pr(\alpha \mid \gamma)\, Pr(\beta \mid \gamma),$$

the right-hand side because $Pr(\beta \mid \gamma) = Pr(\beta \wedge \gamma)/Pr(\gamma) = 0$.
(iii) $Pr(\beta \wedge \gamma) > 0$. Then $Pr(\gamma) \geq Pr(\beta \wedge \gamma) > 0$, so both second disjuncts fail and every conditional probability below is defined. The chain rule inside γ is the bridge:

$$\begin{aligned} Pr(\alpha \wedge \beta \mid \gamma) &= \frac{Pr(\alpha \mid \beta \wedge \gamma)\, Pr(\beta \wedge \gamma)}{Pr(\gamma)} \\ &= Pr(\alpha \mid \beta \wedge \gamma)\, Pr(\beta \mid \gamma). \end{aligned} \tag{$*$}$$

Substituting $Pr(\alpha \mid \beta \wedge \gamma) = Pr(\alpha \mid \gamma)$ into $(*)$ turns (3.15) into (3.16); conversely, (3.16) and $(*)$ give $Pr(\alpha \mid \beta \wedge \gamma) Pr(\beta \mid \gamma) = Pr(\alpha \mid \gamma) Pr(\beta \mid \gamma)$, and $Pr(\beta \mid \gamma) = Pr(\beta \wedge \gamma)/Pr(\gamma) > 0$ cancels, leaving

(3.15).

Problem 3.16. Let X and Y be two binary variables, with values written $x, \bar{x}$ and $y, \bar{y}$ respectively. Show that X and Y are independent if and only if

$$Pr(x, y)\, Pr(\bar{x}, \bar{y}) = Pr(x, \bar{y})\, Pr(\bar{x}, y).$$

Solution. Abbreviate the joint by $a = Pr(x, y)$, $b = Pr(x, \bar{y})$, $c = Pr(\bar{x}, y)$, $d = Pr(\bar{x}, \bar{y})$, so $a+b+c+d = 1$ and case analysis on the eliminated variable gives

$$\begin{aligned} Pr(x) &= a + b, \quad Pr(\bar{x}) = c + d, \\ Pr(y) &= a + c, \quad Pr(\bar{y}) = b + d. \end{aligned}$$

Independence $I_{Pr}(X, \emptyset, Y)$ is, by Equation 3.14, the four equations $Pr(x', y') = Pr(x')Pr(y')$ over $x' \in \{x, \bar{x}\}$, $y' \in \{y, \bar{y}\}$; the claim is that these hold iff $ad = bc$.
($\Rightarrow$) Independence gives $a = Pr(x)Pr(y)$, $b = Pr(x)Pr(\bar{y})$, $c = Pr(\bar{x})Pr(y)$, $d = Pr(\bar{x})Pr(\bar{y})$, whence by commutativity

$$ad = Pr(x)Pr(y)Pr(\bar{x})Pr(\bar{y}) = \big(Pr(x)Pr(\bar{y})\big)\big(Pr(\bar{x})Pr(y)\big) = bc.$$

($\Leftarrow$) Assume $ad = bc$. Expanding and substituting ad for bc,

$$\begin{aligned} Pr(x)\, Pr(y) &= (a + b)(a + c) = a^2 + ac + ab + bc \\ &= a\,(a + b + c + d) = a = Pr(x, y). \end{aligned}$$

The other three cells are the same computation with the same substitution: $(a+b)(b+d) = b(a+c+b+d) = b$, $(c+d)(a+c) = c(a+c+b+d) = c$, and $(c+d)(b+d) = d(a+c+b+d) = d$. (Check!) All four product equations hold, so X and Y are independent.

Problem 3.17. Show that $Pr(\alpha) = O(\alpha)/(1 + O(\alpha))$, where the odds of an event are defined by Equation 3.23,

$$O(\alpha) \stackrel{\text{def}}{=} \frac{Pr(\alpha)}{Pr(\neg\alpha)}.$$

Solution. Write $p = Pr(\alpha)$; Equation 3.23 requires $Pr(\neg\alpha) = 1 - p \neq 0$, so $O(\alpha) = p/(1-p)$ is a finite nonnegative number and $1 + O(\alpha) \neq 0$. Clearing the inner fraction by $1 - p > 0$,

$$\frac{O(\alpha)}{1 + O(\alpha)} = \frac{p/(1-p)}{\big((1-p) + p\big)/(1-p)} = \frac{p}{(1-p) + p} = p = Pr(\alpha).$$

Problem 3.18. Show that

$$\frac{O(\alpha \mid \beta)}{O(\alpha)} = \frac{Pr(\beta \mid \alpha)}{Pr(\beta \mid \neg\alpha)},$$

where $O(\alpha \mid \beta) = Pr(\alpha \mid \beta)/Pr(\neg\alpha \mid \beta)$ is the odds of α after conditioning on β. Note: the quantity $Pr(\beta \mid \alpha)/Pr(\beta \mid \neg\alpha)$ is called the likelihood ratio.

Solution. Rewrite both conditional probabilities in the conditional odds by Bayes rule; assume $Pr(\beta) > 0$, $Pr(\alpha) > 0$ and $Pr(\beta \wedge \neg\alpha) > 0$, which is exactly what makes every quantity in the statement defined and $O(\alpha)$ positive. Then

$$
\begin{aligned}
O(\alpha \mid \beta) &= \frac{Pr(\alpha \mid \beta)}{Pr(\neg\alpha \mid \beta)} \\
&= \frac{Pr(\beta \mid \alpha)\, Pr(\alpha)/Pr(\beta)}{Pr(\beta \mid \neg\alpha)\, Pr(\neg\alpha)/Pr(\beta)} \\
&= \frac{Pr(\beta \mid \alpha)}{Pr(\beta \mid \neg\alpha)} \cdot \frac{Pr(\alpha)}{Pr(\neg\alpha)} \\
&= \frac{Pr(\beta \mid \alpha)}{Pr(\beta \mid \neg\alpha)} \cdot O(\alpha),
\end{aligned}
$$

the common factor $1/Pr(\beta)$ cancelling and the last line using Equation 3.23. Dividing by $O(\alpha) > 0$ gives the claim.

Problem 3.19. Show that events α and β are independent if and only if $O(\alpha \mid \beta) = O(\alpha \mid \neg\beta)$, where $O(\alpha \mid \gamma) = Pr(\alpha \mid \gamma)/Pr(\neg\alpha \mid \gamma)$.

Solution. Assume $0 < Pr(\beta) < 1$ and $Pr(\neg\alpha \mid \beta), Pr(\neg\alpha \mid \neg\beta) > 0$, which is exactly what makes the two conditional odds defined; then the second disjunct of (3.13) is unavailable and independence means $Pr(\alpha \mid \beta) = Pr(\alpha)$.
($\Rightarrow$) From $Pr(\alpha \wedge \beta) = Pr(\alpha)Pr(\beta)$ (Equation 3.14) and case analysis on β,

$$
Pr(\alpha \wedge \neg\beta) = Pr(\alpha) - Pr(\alpha)Pr(\beta) = Pr(\alpha)Pr(\neg\beta),
$$

so dividing by $Pr(\neg\beta) > 0$ gives $Pr(\alpha \mid \neg\beta) = Pr(\alpha) = Pr(\alpha \mid \beta)$. Taking complements within each conditional distribution makes the denominators agree as well, whence

$$
O(\alpha \mid \beta) = \frac{Pr(\alpha)}{1 - Pr(\alpha)} = O(\alpha \mid \neg\beta).
$$

($\Leftarrow$) Let t be the common value of the two odds. Each of $Pr(\cdot \mid \beta)$ and $Pr(\cdot \mid \neg\beta)$ is itself a state of belief, so Exercise 3.17 applies inside each and yields $Pr(\alpha \mid \beta) = t/(1+t) = Pr(\alpha \mid \neg\beta) =: p$. Case analysis on β together with (3.12) then recovers the prior,

$$
Pr(\alpha) = Pr(\alpha \mid \beta)Pr(\beta) + Pr(\alpha \mid \neg\beta)Pr(\neg\beta) = p,
$$

so $Pr(\alpha \mid \beta) = p = Pr(\alpha)$, which is (3.13).

Problem 3.20. Let α and β be two events such that $Pr(\alpha) \neq 0$ and $Pr(\beta) \neq 1$. Suppose that $Pr(\alpha \Rightarrow \beta) = 1$. Show that:
(a) Knowing $\neg\alpha$ will decrease the probability of β, that is, $Pr(\beta \mid \neg\alpha) < Pr(\beta)$.
(b) Knowing β will increase the probability of α, that is, $Pr(\alpha \mid \beta) > Pr(\alpha)$.

Solution. Since $\neg(\alpha \Rightarrow \beta) = \alpha \wedge \neg\beta$, the hypothesis says $Pr(\alpha \wedge \neg\beta) = 0$, so case analysis on β gives

$$
Pr(\alpha) = Pr(\alpha \wedge \beta) + Pr(\alpha \wedge \neg\beta) = Pr(\alpha \wedge \beta), \tag{1}
$$

and monotonicity (Exercise 3.10(c)) applied to $\alpha \wedge \beta \models \beta$ gives

$$
0 < Pr(\alpha) = Pr(\alpha \wedge \beta) \leq Pr(\beta) < 1, \tag{2}
$$

so $Pr(\beta) > 0$ and $Pr(\neg\alpha) > 0$ and both conditionings below are legitimate.
(a) Splitting β on α and using (1), $Pr(\neg\alpha \wedge \beta) = Pr(\beta) - Pr(\alpha)$, so dividing by $Pr(\neg\alpha) = 1 - Pr(\alpha) > 0$,

$$
Pr(\beta \mid \neg\alpha) = \frac{Pr(\beta) - Pr(\alpha)}{1 - Pr(\alpha)} < Pr(\beta),
$$

the inequality because, multiplied through by $1 - Pr(\alpha) > 0$, it reduces to $Pr(\alpha)\big(1 - Pr(\beta)\big) > 0$, which holds by (2).
(b) By (1) and $Pr(\beta) < 1$,

$$Pr(\alpha \mid \beta) = \frac{Pr(\alpha \wedge \beta)}{Pr(\beta)} = \frac{Pr(\alpha)}{Pr(\beta)} > Pr(\alpha).$$

Problem 3.21. Consider Section 3.6.3 and the investigator Rich with his state of belief regarding murder suspects. There is a single variable *Killer* with three values, and Rich's state of belief is:

world	Killer	Pr(.)
ω_1	david	2/3
ω_2	dick	1/6
ω_3	jane	1/6

Of the three suspects, David and Dick are male and Jane is female, so the event "the killer is male" is $\omega_1 \vee \omega_2$.
Suppose now that Rich receives some new evidence that triples his odds of the killer being male. What is the new belief of Rich that David is the killer? What would this belief be if after accommodating the evidence, Rich's belief in the killer being male is 93.75%?

Solution. Both questions have the same answer, $Pr'(Killer = david) = 3/4$.
Write β for $Killer \in \{david, dick\}$ ("the killer is male") and α for $Killer = david$. From the table $Pr(\beta) = 2/3 + 1/6 = 5/6$ and $Pr(\neg\beta) = 1/6$, so $O(\beta) = 5$ by Equation 3.23; and $\alpha \models \beta$, so $Pr(\alpha \wedge \beta) = 2/3$ and $Pr(\alpha \wedge \neg\beta) = 0$.
Bayes factor $k = 3$. Equation 3.25 gives directly

$$Pr'(\alpha) = \frac{k\,Pr(\alpha \wedge \beta) + Pr(\alpha \wedge \neg\beta)}{k\,Pr(\beta) + Pr(\neg\beta)}$$

$$= \frac{3 \cdot \frac{2}{3} + 0}{3 \cdot \frac{5}{6} + \frac{1}{6}} = \frac{2}{\frac{16}{6}} = \frac{3}{4} = 75\%.$$

The same formula, with the same denominator 16/6, supplies the rest of the new state of belief:

world	Killer	Pr(.)	Pr'(.)
ω_1	david	2/3	3/4
ω_2	dick	1/6	3/16
ω_3	jane	1/6	1/16

$Pr'(\beta) = 93.75\%$. This is Jeffrey's rule (Equation 3.21) with $q = 15/16$, $Pr(\alpha \mid \beta) = (2/3)/(5/6) = 4/5$ and $Pr(\alpha \mid \neg\beta) = 0$ (David is male):

$$Pr'(\alpha) = \frac{15}{16} \cdot \frac{4}{5} + \frac{1}{16} \cdot 0 = \frac{3}{4} = 75\%.$$

The two agree because $k = 3$ raises $O(\beta) = 5$ to $O'(\beta) = 15$, i.e. $Pr'(\beta) = 15/16 = 93.75\%$ by Exercise 3.17, so the two specifications describe the same evidence.

2.4 Exercises 3.22–3.27

Problem 3.22. Consider a distribution Pr over variables $\mathbf{X} \cup \{S\}$. Let U be a variable in $\mathbf{X}$ and suppose that S is independent of $\mathbf{X} \setminus \{U\}$ given U. For a given value s of variable S, suppose that $Pr(s|u) = \eta f(u)$ for all values u, where f is some function and $\eta > 0$ is a constant. Show that $Pr(\mathbf{x}|s)$ does not depend on the constant η. That is, $Pr(\mathbf{x}|s)$ is the same for any value of $\eta > 0$ such that $0 \leq \eta f(u) \leq 1$.

Solution. The independence collapses the likelihood of $\mathbf{x}$ to $\eta f(u)$, and the common factor η then cancels against the normalizer. Write $\mathbf{x} = u\mathbf{z}$ with $\mathbf{z}$ the restriction of $\mathbf{x}$ to $\mathbf{Z} = \mathbf{X} \setminus \{U\}$, let $P = Pr(\mathbf{X})$ be

the marginal (which changing η leaves alone, η affecting only the s-row of $Pr(S \mid \mathbf{X})$), and assume $Pr(s) > 0$. The hypothesis $I_{Pr}(S, U, \mathbf{Z})$ is Equation 3.15 at every instantiation with $Pr(u\mathbf{z}) \neq 0$, that is, $Pr(s \mid u\mathbf{z}) = Pr(s \mid u)$, so $Pr(s \mid \mathbf{x}) = \eta f(u)$. By Bayes rule (Equation 3.19) with the denominator expanded by case analysis over the instantiations $\mathbf{x}^\star$ of $\mathbf{X}$,

$$\begin{aligned} Pr(\mathbf{x} \mid s) &= \frac{Pr(s \mid \mathbf{x})\, P(\mathbf{x})}{\sum_{\mathbf{x}^\star} Pr(s \mid \mathbf{x}^\star)\, P(\mathbf{x}^\star)} \\ &= \frac{\eta\, f(u)\, P(\mathbf{x})}{\eta \sum_{\mathbf{x}^\star} f(u^\star)\, P(\mathbf{x}^\star)} = \frac{f(u)\, P(\mathbf{x})}{\sum_{u^\star} f(u^\star)\, P(u^\star)}, \end{aligned}$$

where $u^\star$ is the value $\mathbf{x}^\star$ assigns to U, the last step collects the $\mathbf{x}^\star$ by that value, and cancelling $\eta > 0$ is legitimate because the denominator is $Pr(s) > 0$. The right-hand side mentions only P and f, so it is one and the same number for every admissible η.

Problem 3.23. Prove Equation 3.21. That is, let Pr be a state of belief and let β be an event with $0 < Pr(\beta) < 1$. Let Pr' be the state of belief that results from accommodating soft evidence on β under the "all things considered" method with $Pr'(\beta) = q$, so that Pr' is defined on worlds by Equation 3.20:

$$Pr'(\omega) \stackrel{\text{def}}{=} \begin{cases} \dfrac{q}{Pr(\beta)}\, Pr(\omega), & \text{if } \omega \models \beta \\ \dfrac{1-q}{Pr(\neg\beta)}\, Pr(\omega), & \text{if } \omega \models \neg\beta. \end{cases}$$

Show that for every event α,

$$Pr'(\alpha) = q\, Pr(\alpha|\beta) + (1-q)\, Pr(\alpha|\neg\beta).$$

Solution. Every world satisfying α satisfies exactly one of $\alpha \wedge \beta$ and $\alpha \wedge \neg\beta$, so partitioning $Pr'(\alpha) = \sum_{\omega \models \alpha} Pr'(\omega)$ (Equation 3.1) accordingly and substituting Equation 3.20 in each part,

$$\begin{aligned} Pr'(\alpha) &= \frac{q}{Pr(\beta)} \sum_{\omega \models \alpha \wedge \beta} Pr(\omega) + \frac{1-q}{Pr(\neg\beta)} \sum_{\omega \models \alpha \wedge \neg\beta} Pr(\omega) \\ &= \frac{q}{Pr(\beta)}\, Pr(\alpha \wedge \beta) + \frac{1-q}{Pr(\neg\beta)}\, Pr(\alpha \wedge \neg\beta) \\ &= q\, Pr(\alpha|\beta) + (1-q)\, Pr(\alpha|\neg\beta), \end{aligned}$$

the second line by Equation 3.1 again, applied to $\alpha \wedge \beta$ and $\alpha \wedge \neg\beta$, and the third by Bayes conditioning (Equation 3.12), both quotients being defined since $0 < Pr(\beta) < 1$. This is Equation 3.21.

Problem 3.24. Prove Equations 3.24 and 3.25. That is, recall that the odds of an event β are defined by Equation 3.23 as

$$O(\beta) \stackrel{\text{def}}{=} \frac{Pr(\beta)}{Pr(\neg\beta)},$$

and that soft evidence on β may be specified by the Bayes factor

$$k = \frac{O'(\beta)}{O(\beta)}, \qquad O'(\beta) = \frac{Pr'(\beta)}{Pr'(\neg\beta)},$$

where Pr' is the state of belief after the evidence is accommodated. Assuming $0 < Pr(\beta) < 1$ and $k > 0$, show that

$$Pr'(\beta) = \frac{k\, Pr(\beta)}{k\, Pr(\beta) + Pr(\neg\beta)}$$

(this is Equation 3.24) and that, if Pr' is then obtained from Pr by Jeffrey's rule (Equation 3.21) with $Pr'(\beta) = q$ given by 3.24, then for every event α

$$Pr'(\alpha) = \frac{k\,Pr(\alpha \wedge \beta) + Pr(\alpha \wedge \neg\beta)}{k\,Pr(\beta) + Pr(\neg\beta)},$$

which is Equation 3.25.

Solution. Equation 3.24. With $q = Pr'(\beta)$ the constraint $O'(\beta) = k\,O(\beta)$ reads $q/(1-q) = k\,Pr(\beta)/Pr(\neg\beta)$, whose right-hand side is finite because $Pr(\neg\beta) > 0$; hence $q \neq 1$ and cross-multiplying is legitimate:

$$q\,Pr(\neg\beta) = k\,Pr(\beta)\,(1-q), \qquad \text{i.e.} \qquad q\,\big(k\,Pr(\beta) + Pr(\neg\beta)\big) = k\,Pr(\beta).$$

The coefficient is strictly positive ($k > 0$ and $0 < Pr(\beta) < 1$), so dividing by it gives Equation 3.24, and correspondingly

$$1 - q \;=\; \frac{Pr(\neg\beta)}{k\,Pr(\beta) + Pr(\neg\beta)}.$$

Equation 3.25. Abbreviate $D = k\,Pr(\beta) + Pr(\neg\beta) > 0$. Jeffrey's rule (Equation 3.21) with these q and $1-q$, together with Bayes conditioning (Equation 3.12), gives

$$\begin{aligned} Pr'(\alpha) &= q\,Pr(\alpha|\beta) + (1-q)\,Pr(\alpha|\neg\beta) \\ &= \frac{k\,Pr(\beta)}{D} \cdot \frac{Pr(\alpha \wedge \beta)}{Pr(\beta)} + \frac{Pr(\neg\beta)}{D} \cdot \frac{Pr(\alpha \wedge \neg\beta)}{Pr(\neg\beta)} \\ &= \frac{k\,Pr(\alpha \wedge \beta) + Pr(\alpha \wedge \neg\beta)}{k\,Pr(\beta) + Pr(\neg\beta)}, \end{aligned}$$

the cancellations being legitimate because $Pr(\beta)$ and $Pr(\neg\beta)$ are nonzero.
For the burglary state of belief of Section 3.6.2,

world	Alarm	Burglary	$Pr(.)$
ω_1	true	true	.000095
ω_2	true	false	.009999
ω_3	false	true	.000005
ω_4	false	false	.989901

we have $Pr(Alarm) = .010094$ and $Pr(\neg Alarm) = .989906$, so $\alpha : Burglary$, $\beta : Alarm$ and $k = 4$ in 3.25 give

$$Pr'(Burglary) = \frac{4(.000095) + .000005}{4(.010094) + .989906} = \frac{.000385}{1.030282} \approx 3.74 \times 10^{-4}.$$

Problem 3.25. Suppose we transmit a bit across a noisy channel but for bit 0 we send a signal -1 and for bit 1 we send a signal $+1$. Suppose again that Gaussian noise is added to the reading y from the noisy channel, with densities

$$\begin{aligned} f(y|X=0) &= \frac{1}{\sqrt{2\pi\sigma^2}}\, e^{-(y+1)^2/2\sigma^2} \\ f(y|X=1) &= \frac{1}{\sqrt{2\pi\sigma^2}}\, e^{-(y-1)^2/2\sigma^2}. \end{aligned}$$

(a) Show that if we treat the reading y of a continuous variable Y as soft evidence on $X = 0$, the corresponding Bayes factor is

$$k = e^{-2y/\sigma^2}.$$

(b) Give the corresponding Bayes factors for the following readings y and standard deviations σ:

(i) $y = +\frac{1}{2}$ and $\sigma = \frac{1}{4}$
(ii) $y = -\frac{1}{2}$ and $\sigma = \frac{1}{4}$
(iii) $y = -\frac{3}{2}$ and $\sigma = \frac{4}{5}$
(iv) $y = +\frac{1}{4}$ and $\sigma = \frac{4}{5}$
(v) $y = -1$ and $\sigma = 2$
(c) What reading y would result in neutral evidence regardless of the standard deviation? What reading y would result in a Bayes factor of 2 given a standard deviation $\sigma = 0.2$?

Solution. (a) By Equation 3.28 the Bayes factor of the reading y as soft evidence on $X = 0$ is the likelihood ratio $f(y|X{=}0)/f(y|X{=}1)$. The identical normalizers $1/\sqrt{2\pi\sigma^2}$ cancel and $-(y+1)^2 + (y-1)^2 = -4y$, so

$$k = e^{[-(y+1)^2+(y-1)^2]/2\sigma^2} = e^{-4y/2\sigma^2} = e^{-2y/\sigma^2}.$$

(b) Evaluating $k = e^{-2y/\sigma^2}$ in each case:

case	y	σ	exponent $-2y/\sigma^2$	k	k (numeric)
(i)	$+1/2$	$1/4$	-16	e^{-16}	1.13×10^{-7}
(ii)	$-1/2$	$1/4$	$+16$	e^{16}	8.89×10^{6}
(iii)	$-3/2$	$4/5$	$+75/16$	$e^{75/16}$	108.6
(iv)	$+1/4$	$4/5$	$-25/32$	$e^{-25/32}$	0.458
(v)	-1	2	$+1/2$	$\sqrt{e}$	1.649

(c) The reading $y = 0$, and only that reading, is neutral for every σ:

$$e^{-2y/\sigma^2} = 1 \iff -\frac{2y}{\sigma^2} = 0 \iff y = 0,$$

since $\sigma^2 > 0$. For a Bayes factor of 2 with $\sigma = 0.2$, so $\sigma^2 = 0.04$,

$$e^{-50y} = 2 \iff -50y = \ln 2 \iff y = -\frac{\ln 2}{50} \approx -0.0139.$$

Problem 3.26. Prove Equation 3.27. That is, let X be a binary variable with values $\{x, \bar{x}\}$ and let Y be a continuous variable with values $y \in (-\infty, +\infty)$, where the conditional densities of Y given the two values of X are $f(y|x)$ and $f(y|\bar{x})$. Show that

$$\frac{Pr(x|y)/Pr(\bar{x}|y)}{Pr(x)/Pr(\bar{x})} = \frac{f(y|x)}{f(y|\bar{x})}.$$

Hint: Show first that $Pr(x|y)/Pr(x) = f(y|x)/f(y)$, where $f(y)$ is the PDF for variable Y.

Solution. Since $Y = y$ typically has probability zero, $Pr(x|y)$ means the limit of conditioning on a shrinking interval, $Pr(x|y) = \lim_{\Delta\to 0^+} Pr(x \mid A_\Delta)$ with A_Δ the event $y \le Y \le y + \Delta$. Assume $0 < Pr(x) < 1$, that $f(\cdot|x)$ and $f(\cdot|\bar{x})$ are continuous at y, and that $f(y) > 0$ and $f(y|\bar{x}) > 0$.
Case analysis on X applied to the event $Y \le t$ gives $F(t) = F(t|x)Pr(x) + F(t|\bar{x})Pr(\bar{x})$ for the CDFs, so differentiating (each CDF is the integral of its density, Section 3.7.1),

$$f(t) = f(t|x)\,Pr(x) + f(t|\bar{x})\,Pr(\bar{x}),$$

whence f is continuous at y too. Bayes rule (Equation 3.19) applied to the ordinary event A_Δ — legitimate for all small $\Delta > 0$, since $Pr(A_\Delta) \ge \Delta \cdot \min_{[y,y+\Delta]} f > 0$ by continuity and positivity of f at y — gives, after dividing by $Pr(x) > 0$ and writing each probability as an integral scaled by Δ,

$$\frac{Pr(x \mid A_\Delta)}{Pr(x)} = \frac{Pr(A_\Delta \mid x)}{Pr(A_\Delta)} = \frac{\frac{1}{\Delta}\int_y^{y+\Delta} f(t|x)\,dt}{\frac{1}{\Delta}\int_y^{y+\Delta} f(t)\,dt}.$$

Each average lies between the minimum and the maximum of a function continuous at y, so it tends

to that function's value at y as $\Delta \to 0^+$; the quotient therefore tends to $f(y|x)/f(y)$, and the left side to $Pr(x|y)/Pr(x)$, which is the hint:

$$\frac{Pr(x|y)}{Pr(x)} = \frac{f(y|x)}{f(y)}.$$

The identical argument with $\bar{x}$ in place of x gives $Pr(\bar{x}|y)/Pr(\bar{x}) = f(y|\bar{x})/f(y)$, which is nonzero; dividing the two identities cancels $f(y)$, and rearranging the four factors on the left,

$$\frac{Pr(x|y)/Pr(x)}{Pr(\bar{x}|y)/Pr(\bar{x})} = \frac{Pr(x|y)\,Pr(\bar{x})}{Pr(\bar{x}|y)\,Pr(x)} = \frac{Pr(x|y)/Pr(\bar{x}|y)}{Pr(x)/Pr(\bar{x})} = \frac{f(y|x)}{f(y|\bar{x})},$$

which is Equation 3.27.
Method (2): apply Bayes rule to A_Δ at both values of X and take the ratio, so that $Pr(A_\Delta)$ cancels at once:

$$\frac{Pr(x|A_\Delta)}{Pr(\bar{x}|A_\Delta)} = \frac{Pr(A_\Delta|x)\,Pr(x)}{Pr(A_\Delta|\bar{x})\,Pr(\bar{x})};$$

dividing by $Pr(x)/Pr(\bar{x})$ and letting $\Delta \to 0^+$ gives 3.27 without the marginal density.

Problem 3.27. Suppose we have a sensor that bears on event β and has a false positive rate f_p and a false negative rate f_n. Suppose further that we want a positive reading of this sensor to increase the odds of β by a factor of $k > 1$ and a negative reading to decrease the odds of β by the same factor k. Prove that these conditions imply that $f_p = f_n = 1/(k+1)$.

Solution. Let S be the event that the sensor reads positive, so $f_p = Pr(S|\neg\beta)$ and $f_n = Pr(\neg S|\beta)$, whence $Pr(S|\beta) = 1 - f_n$ and $Pr(\neg S|\neg\beta) = 1 - f_p$ by Equation 3.5; assume $0 < Pr(\beta) < 1$ and $0 < f_p < 1$, which is exactly what makes $O(\beta)$ positive and both Bayes factors finite. Bayes rule (Equation 3.19) applied to numerator and denominator of the posterior odds, with the common factor $Pr(S)$ cancelling, gives

$$O'(\beta) = \frac{Pr(S|\beta)\,Pr(\beta)}{Pr(S|\neg\beta)\,Pr(\neg\beta)} = \frac{1 - f_n}{f_p}\,O(\beta),$$

so $k^+ = (1-f_n)/f_p$; the same computation with $\neg S$ for S gives $k^- = f_n/(1-f_p)$. The two requirements are $k^+ = k$ and $k^- = 1/k$ (a decrease by a factor k divides the odds by k), that is, clearing the nonzero denominators,

$$1 - f_n = k\,f_p, \qquad k\,f_n = 1 - f_p.$$

Substituting $f_p = 1 - kf_n$ from the second into the first,

$$1 - f_n = k\,(1 - kf_n) \iff (k^2 - 1)\,f_n = k - 1 \iff (k-1)(k+1)\,f_n = k - 1,$$

and $k > 1$ permits cancelling $k - 1$, leaving $f_n = 1/(k+1)$ and hence

$$f_p = 1 - kf_n = 1 - \frac{k}{k+1} = \frac{1}{k+1} = f_n.$$

3 Bayesian Networks

Contents

3.1 Exercises 4.1–4.7 29
3.2 Exercises 4.8–4.14 34
3.3 Exercises 4.15–4.21 39
3.4 Exercises 4.22–4.25 44

3.1 Exercises 4.1–4.7

Problem 4.1. Consider the Bayesian network of Figure 4.14. Its DAG G has eight binary variables A, B, C, D, E, F, G, H and the following edges:

$$A \to C, \quad A \to D, \quad B \to D, \quad B \to E,$$
$$C \to F, \quad D \to F, \quad F \to G, \quad F \to H, \quad E \to H.$$

So A and B are roots; C has parent A; D has parents A, B; E has parent B; F has parents C, D; G has parent F; and H has parents E, F. Some of the CPTs are given:

A	Θ_A
1	.2
0	.8

B	Θ_B
1	.7
0	.3

B	E	$\Theta_{E\|B}$
1	1	.1
1	0	.9
0	1	.9
0	0	.1

A	B	D	$\Theta_{D\|AB}$
1	1	1	.5
1	1	0	.5
1	0	1	.6
1	0	0	.4
0	1	1	.1
0	1	0	.9
0	0	1	.8
0	0	0	.2

- (a) List the Markovian assumptions asserted by the DAG.
- (b) Express $\Pr(a, b, c, d, e, f, g, h)$ in terms of network parameters.
- (c) Compute $\Pr(A = 0, B = 0)$ and $\Pr(E = 1 \mid A = 1)$. Justify your answers.
- (d) True or false? Why?
  - $\text{dsep}(A, BH, E)$
  - $\text{dsep}(G, D, E)$
  - $\text{dsep}(AB, F, GH)$

Solution. Part (a). By (4.1), Markov(G) has one statement $I(V, \text{Parents}(V), \text{NonDescendants}(V))$ per variable V; reading the descendants off the DAG, the Markovian assumptions are

$$I(A, \emptyset, \{B, E\}), \qquad I(B, \emptyset, \{A, C\}),$$
$$I(C, A, \{B, D, E\}), \qquad I(D, AB, \{C, E\}),$$
$$I(E, B, \{A, C, D, F, G\}), \qquad I(F, CD, \{A, B, E\}),$$
$$I(G, F, \{A, B, C, D, E, H\}), \qquad I(H, EF, \{A, B, C, D, G\}).$$

Part (b). By the chain rule for Bayesian networks (4.2), each variable contributes exactly one factor, its own parameter given the values its parents receive:

$$\Pr(a, b, c, d, e, f, g, h) = \theta_a \, \theta_b \, \theta_{c|a} \, \theta_{d|ab} \, \theta_{e|b} \, \theta_{f|cd} \, \theta_{g|f} \, \theta_{h|ef}.$$

Part (c). Both quantities are decided by independencies the DAG guarantees, so the incomplete parametrization is no obstacle.

$\Pr(A=0,B=0)$. Since $I(B,\emptyset,\{A,C\})$ lies in Markov(G), decomposition (4.4) yields $I_{\Pr}(B,\emptyset,A)$ (equivalently, A,B are distinct roots, so Exercise 4.3 and Theorem 4.2 apply), whence

$$\Pr(A=0,B=0)=\theta_{A=0}\,\theta_{B=0}=(.8)(.3)=.24.$$

$\Pr(E=1\mid A=1)$. Here $\mathrm{dsep}_G(A,\emptyset,E)$ holds: with an empty conditioning set a convergent valve is automatically closed, and each path from A to E carries one (the neighbours of E are only B and H, so every path ends $B\to E$ or $H\leftarrow E$):

$$\begin{aligned}
&A\to D\leftarrow B\to E && \text{closed at } D,\\
&A\to C\to F\leftarrow D\leftarrow B\to E && \text{closed at } F,\\
&A\to C\to F\to H\leftarrow E && \text{closed at } H,\\
&A\to D\to F\to H\leftarrow E && \text{closed at } H,
\end{aligned}$$

so every path is blocked. By Theorem 4.2, $I_{\Pr}(A,\emptyset,E)$, whence

$$\Pr(E=1\mid A=1)=\Pr(E=1)=\sum_b \theta_{E=1|b}\,\theta_b=(.1)(.7)+(.9)(.3)=.34.$$

Part (d). All three are **false**; one open path each.
(i) $\mathrm{dsep}(A,\{B,H\},E)$: the path $A\to C\to F\to H\leftarrow E$ is sequential at C and at F, neither in $\{B,H\}$, and convergent at $H\in\{B,H\}$, so every valve is open.
(ii) $\mathrm{dsep}(G,D,E)$: on the path

$$G\leftarrow F\leftarrow C\leftarrow A\to D\leftarrow B\to E$$

the valves are sequential at F and C, divergent at A and at B, none of them D, and convergent at the evidence node D — all open.
(iii) $\mathrm{dsep}(AB,F,GH)$: the path $B\to E\to H$ joins $B\in\{A,B\}$ to $H\in\{G,H\}$ through its single valve, sequential at $E\notin\{F\}$, hence open.

Problem 4.2. Consider the DAG G in Figure 4.15. It has nine nodes arranged in a 3×3 grid, $A_1A_2A_3$ on top, $B_1B_2B_3$ in the middle and $C_1C_2C_3$ at the bottom, with edges

$$\begin{aligned}
&A_1\to A_2,\quad A_2\to A_3,\quad A_1\to B_2,\quad A_2\to B_3,\\
&B_1\to B_2,\quad B_2\to B_3,\quad B_1\to C_2,\quad B_2\to C_3,\\
&C_1\to C_2,\quad C_2\to C_3.
\end{aligned}$$

Equivalently, the parent sets are $\mathrm{Parents}(A_1)=\mathrm{Parents}(B_1)=\mathrm{Parents}(C_1)=\emptyset$, $\mathrm{Parents}(A_2)=\{A_1\}$, $\mathrm{Parents}(B_2)=\{A_1,B_1\}$, $\mathrm{Parents}(C_2)=\{B_1,C_1\}$, $\mathrm{Parents}(A_3)=\{A_2\}$, $\mathrm{Parents}(B_3)=\{A_2,B_2\}$, $\mathrm{Parents}(C_3)=\{B_2,C_2\}$.
Determine if any of $\mathrm{dsep}_G(A_i,\emptyset,B_i)$, $\mathrm{dsep}_G(A_i,\emptyset,C_i)$, or $\mathrm{dsep}_G(B_i,\emptyset,C_i)$ hold for $i=1,2,3$.

Solution. With an empty conditioning set every convergent valve is closed and every sequential or divergent valve open, so an unblocked path is exactly one that climbs from one endpoint to some node W and descends to the other. Hence

$$\mathrm{dsep}_G(X,\emptyset,Y)\text{ fails}\iff \mathrm{Anc}(X)\cap\mathrm{Anc}(Y)\neq\emptyset,$$

where $\mathrm{Anc}(V)$ is V together with its ancestors. (Left to right, the turning point W is a common ancestor. Right to left, pick a common ancestor W minimising the total length of shortest directed paths $W\to\cdots\to X$ and $W\to\cdots\to Y$; these share no node but W, since a shared node would be a common ancestor with a smaller total, so splicing them gives a path all of whose valves are sequential or divergent.)
The ancestral sets in this DAG are

$$\begin{aligned}
&\mathrm{Anc}(A_1) = \{A_1\}, && \mathrm{Anc}(B_1) = \{B_1\},\\
&\mathrm{Anc}(C_1) = \{C_1\}, && \mathrm{Anc}(A_2) = \{A_1, A_2\},\\
&\mathrm{Anc}(B_2) = \{A_1, B_1, B_2\}, && \mathrm{Anc}(C_2) = \{B_1, C_1, C_2\},\\
&\mathrm{Anc}(A_3) = \{A_1, A_2, A_3\},\\
&\mathrm{Anc}(B_3) = \{A_1, A_2, B_1, B_2, B_3\},\\
&\mathrm{Anc}(C_3) = \{A_1, B_1, B_2, C_1, C_2, C_3\}.
\end{aligned}$$

Case $i = 1$. All three **hold**: A_1, B_1, C_1 are roots, so their ancestral sets are the pairwise disjoint singletons $\{A_1\}, \{B_1\}, \{C_1\}$ (also immediate from Exercise 4.3).
Case $i = 2$. $\mathrm{dsep}_G(A_2, \emptyset, B_2)$ **fails**: A_1 is a common ancestor, witnessing the unblocked path $A_2 \leftarrow A_1 \rightarrow B_2$, divergent at A_1. Likewise $\mathrm{dsep}_G(B_2, \emptyset, C_2)$ **fails** on the common ancestor B_1 and the path $B_2 \leftarrow B_1 \rightarrow C_2$. But $\mathrm{dsep}_G(A_2, \emptyset, C_2)$ **holds**, since $\mathrm{Anc}(A_2) = \{A_1, A_2\}$ and $\mathrm{Anc}(C_2) = \{B_1, C_1, C_2\}$ are disjoint; explicitly, each of the paths

$$\begin{aligned}
&A_2 \leftarrow A_1 \rightarrow B_2 \leftarrow B_1 \rightarrow C_2 && \text{closed at } B_2,\\
&A_2 \leftarrow A_1 \rightarrow B_2 \rightarrow C_3 \leftarrow C_2 && \text{closed at } C_3,\\
&A_2 \rightarrow B_3 \leftarrow B_2 \leftarrow B_1 \rightarrow C_2 && \text{closed at } B_3,\\
&A_2 \rightarrow B_3 \leftarrow B_2 \rightarrow C_3 \leftarrow C_2 && \text{closed at } B_3,
\end{aligned}$$

contains a convergent valve, which is closed given $\emptyset$.
Case $i = 3$. All three **fail**, on the common ancestors A_2, A_1 and B_2 respectively, which give the unblocked paths

$$\begin{aligned}
&A_3 \leftarrow A_2 \rightarrow B_3,\\
&A_3 \leftarrow A_2 \leftarrow A_1 \rightarrow B_2 \rightarrow C_3,\\
&B_3 \leftarrow B_2 \rightarrow C_3,
\end{aligned}$$

whose valves are all sequential or divergent, hence open.
Summary. Of the nine statements exactly four hold: all three at $i = 1$, plus $\mathrm{dsep}_G(A_2, \emptyset, C_2)$.

Problem 4.3. Show that every root variable X in a DAG G is d-separated from every other root variable Y. (A root variable is one with no parents; the claim is that $\mathrm{dsep}_G(X, \emptyset, Y)$ holds.)

Solution. Every path between two distinct roots contains a convergent valve, which is closed given $\emptyset$. Let

$$\pi : \; X = N_0 - N_1 - \cdots - N_k = Y$$

be any path between roots $X \neq Y$, and let $d_j \in \{\rightarrow, \leftarrow\}$ record the orientation of the j-th edge as traversed from N_0. Then $k \geq 2$: $k \geq 1$ as $X \neq Y$, and $k = 1$ would make one root a parent of the other. Moreover $d_1 = \rightarrow$ and $d_k = \leftarrow$, since no edge points into a root. Take the smallest i with $d_{i+1} = \leftarrow$, well defined because $d_k = \leftarrow$; then $d_i = \rightarrow$ by minimality (for $i = 1$ this is d_1), so π contains

$$N_{i-1} \rightarrow N_i \leftarrow N_{i+1},$$

a convergent valve at N_i, closed by Definition 4.2 because $\mathbf{Z} = \emptyset$ contains neither N_i nor any of its descendants. Hence every path is blocked and $\mathrm{dsep}_G(X, \emptyset, Y)$.

Problem 4.4. Consider a Bayesian network over variables $\mathbf{X}, S$ that induces a distribution Pr. Suppose that S is a leaf node in the network that has a single parent $U \in \mathbf{X}$. For a given value s of variable S, show that $\Pr(\mathbf{x} \mid s)$ does not change if we change the CPT of variable S as follows:

$$\theta'_{s|u} = \eta\,\theta_{s|u}$$

for all u and some constant $\eta > 0$.

Solution. Because S is a leaf it belongs to no family but its own, so its parameter factors out of the chain rule (4.2) and the rescaling becomes a common factor. Write $u(\mathbf{x})$ for the value $\mathbf{x}$ assigns to U and

$$f(\mathbf{x}) \;=\; \prod_{X \in \mathbf{X}} \theta_{x|\mathbf{u}_X}$$

for the product of the parameters contributed by the families of $\mathbf{X}$, none of which mentions S. Then (4.2) gives

$$\Pr(\mathbf{x}, s) = \theta_{s|u(\mathbf{x})}\, f(\mathbf{x}), \qquad \Pr'(\mathbf{x}, s) = \eta\,\theta_{s|u(\mathbf{x})}\, f(\mathbf{x}) = \eta\,\Pr(\mathbf{x}, s),$$

with the **same** constant η for every $\mathbf{x}$, precisely because η does not depend on u. (Only the s-row of $\Theta_{S|U}$ is constrained; the other rows may be set to restore normalisation, and the argument never touches them.) Summing over the instantiations $\mathbf{x}$ gives $\Pr'(s) = \eta\,\Pr(s)$, so assuming $\Pr(s) > 0$ — needed for $\Pr(\mathbf{x} \mid s)$ to be defined, and giving $\Pr'(s) > 0$ since $\eta > 0$ —

$$\Pr'(\mathbf{x} \mid s) = \frac{\eta\,\Pr(\mathbf{x}, s)}{\eta\,\Pr(s)} = \Pr(\mathbf{x} \mid s).$$

Problem 4.5. Consider the distribution Pr defined by Equation 4.2 and DAG G. That is, G is a DAG over variables $\mathbf{Z}$, each variable X with parents $\mathbf{U}$ carries a CPT $\Theta_{X|\mathbf{U}}$ with entries $\theta_{x|\mathbf{u}} \geq 0$ satisfying $\sum_x \theta_{x|\mathbf{u}} = 1$ for every parent instantiation $\mathbf{u}$, and

$$\Pr(\mathbf{z}) \;\stackrel{\text{def}}{=} \prod_{\theta_{x|\mathbf{u}} \sim \mathbf{z}} \theta_{x|\mathbf{u}}.$$

Show the following:

- (a) $\sum_{\mathbf{z}} \Pr(\mathbf{z}) = 1$.
- (b) Pr satisfies the independencies in Markov(G).
- (c) $\Pr(x \mid \mathbf{u}) = \theta_{x|\mathbf{u}}$ for every value x of variable X and every instantiation $\mathbf{u}$ of its parents $\mathbf{U}$.

Solution. Everything follows from one lemma about prefixes of a topological order. Fix a topological ordering $X_1, \ldots, X_n$ of $\mathbf{Z}$ (it exists since G is acyclic), so the parents $\mathbf{U}_i$ of X_i occur among $X_1, \ldots, X_{i-1}$; let $\mathbf{u}_i$ be the value an instantiation assigns to $\mathbf{U}_i$. Each variable contributes exactly one compatible parameter, so (4.2) reads

$$\Pr(x_1, \ldots, x_n) = \prod_{i=1}^{n} \theta_{x_i|\mathbf{u}_i}. \qquad (*)$$

Lemma. For every $0 \leq k \leq n$ and every instantiation $x_1 \cdots x_k$,

$$\Pr(x_1, \ldots, x_k) = \sum_{x_{k+1}, \ldots, x_n} \Pr(x_1, \ldots, x_n) = \prod_{i=1}^{k} \theta_{x_i|\mathbf{u}_i}. \qquad (\dagger)$$

Downward induction on k: at $k = n$ this is $(*)$, and summing $(\dagger)$ at $k+1$ over the values of X_{k+1} factors out $\prod_{i \leq k} \theta_{x_i|\mathbf{u}_i}$ — legitimately, since $\mathbf{U}_{k+1} \subseteq \{X_1, \ldots, X_k\}$ makes $\mathbf{u}_{k+1}$ a function of $x_1 \cdots x_k$ alone — and leaves $\sum_x \theta_{x|\mathbf{u}_{k+1}} = 1$ by Definition 4.1.

Part (a). Take $k = 0$: the empty product is 1, so $\sum_{\mathbf{z}} \Pr(\mathbf{z}) = 1$; and $\Pr(\mathbf{z}) \geq 0$ since every parameter is nonnegative.

Part (c). For $X = X_i$ with parents $\mathbf{U}$, applying $(\dagger)$ at $k = i$ and at $k = i - 1$ gives, for every $x_1 \cdots x_{i-1}$ assigning $\mathbf{u}$ to $\mathbf{U}$,

$$\Pr(x_1, \ldots, x_{i-1}, x) = \Pr(x_1, \ldots, x_{i-1})\, \theta_{x|\mathbf{u}},$$

the factor $\theta_{x|\mathbf{u}}$ depending on $x_1 \cdots x_{i-1}$ only through $\mathbf{u}$. Summing over the instantiations $\mathbf{v}$ of $\mathbf{V} = \{X_1, \ldots, X_{i-1}\} \setminus \mathbf{U}$,

$$\Pr(x, \mathbf{u}) = \theta_{x|\mathbf{u}} \sum_{\mathbf{v}} \Pr(\mathbf{v}, \mathbf{u}) = \theta_{x|\mathbf{u}}\ \Pr(\mathbf{u}),$$

so $\Pr(x \mid \mathbf{u}) = \theta_{x|\mathbf{u}}$ whenever $\Pr(\mathbf{u}) > 0$ (otherwise the claim is vacuous).
Part (b). The lemma holds for **every** topological order, so choose one adapted to X. Put $\mathbf{D} = \{X\} \cup \text{Descendants}(X)$, so that $\mathbf{Z} \setminus \mathbf{D} = \mathbf{U} \cup \mathbf{N}$ with $\mathbf{N} = \text{NonDescendants}(X)$; this set is closed under parents, since a parent in $\mathbf{D}$ would make its child a descendant of X. Hence listing $\mathbf{Z} \setminus \mathbf{D}$ first, then X, then $\mathbf{D} \setminus \{X\}$, each block in a topological order of the sub-DAG it induces, is a topological order of G. With it $X = X_i$ and $\{X_1, \ldots, X_{i-1}\} = \mathbf{U} \cup \mathbf{N}$, so (†) at $k = i$ and $k = i - 1$ gives

$$\Pr(x, \mathbf{u}, \mathbf{n}) = \Pr(\mathbf{u}, \mathbf{n})\, \theta_{x|\mathbf{u}}.$$

When $\Pr(\mathbf{u}, \mathbf{n}) > 0$, dividing gives $\Pr(x \mid \mathbf{u}, \mathbf{n}) = \theta_{x|\mathbf{u}}$, and $\Pr(\mathbf{u}) \geq \Pr(\mathbf{u}, \mathbf{n}) > 0$ lets part (c) rewrite the right-hand side as $\Pr(x \mid \mathbf{u})$; the instantiations with $\Pr(\mathbf{u}, \mathbf{n}) = 0$ are excused by the definition of $I_{\Pr}$ in Section 4.4. Hence $I_{\Pr}(X, \mathbf{U}, \mathbf{N})$ for every X, which is Markov(G).

Problem 4.6. Use the graphoid axioms to prove $\text{dsep}_G(S_1, S_2, \{S_3, \ldots, S_n\})$ in the DAG G of Figure 4.11. Assume that you are given the Markovian assumptions for DAG G.
Figure 4.11 is the hidden Markov model structure: it has state variables $S_1, S_2, \ldots, S_n$ forming a chain

$$S_1 \to S_2 \to S_3 \to \cdots \to S_n,$$

together with an observation variable O_i for each i, with the single edge $S_i \to O_i$. Thus $\text{Parents}(S_1) = \emptyset$, $\text{Parents}(S_i) = \{S_{i-1}\}$ for $i \geq 2$, and $\text{Parents}(O_i) = \{S_i\}$ for all i. Assume $n \geq 3$.

Solution. Induct on k to get $(\text{C}_k) = I(S_1, S_2, \{S_3, \ldots, S_k\})$ for $3 \leq k \leq n$; at $k = n$ this is the independence that $\text{dsep}_G(S_1, S_2, \{S_3, \ldots, S_n\})$ asserts.
The descendants of S_i are $S_{i+1}, \ldots, S_n$ and $O_i, \ldots, O_n$, so for $i \geq 2$ the set Markov(G) contains $I(S_i, S_{i-1}, \{S_1, \ldots, S_{i-2}\} \cup \{O_1, \ldots, O_{i-1}\})$, and decomposition (4.4) discards the observation variables, leaving

$$I\big(S_i,\ S_{i-1},\ \{S_1, \ldots, S_{i-2}\}\big). \qquad (\text{N}_i)$$

Base $k = 3$: (N_3) is $I(S_3, S_2, \{S_1\})$, so symmetry (4.3) gives (C_3).
Step: let $3 \leq k < n$ and assume (C_k). Weak union (4.7) applied to (N_{k+1}) with $\mathbf{Z} = \{S_k\}$, $\mathbf{Y} = \{S_2, \ldots, S_{k-1}\}$, $\mathbf{W} = \{S_1\}$, followed by symmetry (4.3), yields

$$I\big(S_1,\ \{S_2, \ldots, S_k\},\ S_{k+1}\big).$$

Contraction (4.9) on (C_k) and this statement, with $\mathbf{Z} = \{S_2\}$, $\mathbf{Y} = \{S_3, \ldots, S_k\}$ (so $\mathbf{Z} \cup \mathbf{Y} = \{S_2, \ldots, S_k\}$) and $\mathbf{W} = \{S_{k+1}\}$, gives (C_{k+1}).

Problem 4.7. Show that $\text{dsep}_G(\mathbf{X}, \mathbf{Z}, \mathbf{Y})$ can be decided in time and space that are linear in the size of DAG G based on Theorem 4.1.
Recall Theorem 4.1: testing whether $\mathbf{X}$ and $\mathbf{Y}$ are d-separated by $\mathbf{Z}$ in DAG G is equivalent to testing whether $\mathbf{X}$ and $\mathbf{Y}$ are disconnected in the DAG G' obtained by pruning G as follows: repeatedly delete any leaf node W that does not belong to $\mathbf{X} \cup \mathbf{Y} \cup \mathbf{Z}$, until no more nodes can be deleted; then delete all edges outgoing from nodes in $\mathbf{Z}$. The connectivity test on G' ignores edge directions.

Solution. Store G, with n nodes and e edges, in adjacency-list form with each node holding its child list and its parent list (either is computable from the other in $O(n+e)$); the three steps below run in $O(n+e)$ time and $O(n)$ auxiliary space, which is linear in the size $\Theta(n+e)$ of G.
Preprocessing, $O(n+e)$: boolean arrays marked (true exactly on $\mathbf{X}\cup\mathbf{Y}\cup\mathbf{Z}$), inZ, inY, deleted (initially false), and an integer array outdeg$[V]$ holding the number of children of V.
Step 1, prune leaves to a fixed point. A node is a leaf of the current graph iff outdeg $= 0$. Initialise a work list Q with every unmarked node of out-degree 0; while Q is nonempty, pop an undeleted W, set deleted$[W]$, and for each undeleted parent P of W decrement outdeg$[P]$, pushing P if that makes it an unmarked leaf. Since outdeg$[V]$ is invariantly the number of undeleted children of V, a node enters Q exactly when it first becomes an unmarked leaf, so the loop halts precisely at the fixed point Theorem 4.1 demands; out-degrees only decrease, so each node is pushed once and each parent list scanned once, giving $O(n+e)$.
Step 2, delete the edges outgoing from $\mathbf{Z}$: no change to the data structure is needed. Declare an edge $P \to C$ usable iff neither endpoint is deleted and inZ$[P]$ = **false**, and let Step 3 cross usable edges in either direction. The test must be on the **tail** P even when the search sits at C, since deleting $P \to C$ makes it unusable both ways: in $A \to P \to C$ with $\mathbf{Z} = \{P\}$, $\mathbf{X} = \{C\}$, $\mathbf{Y} = \{A\}$, the graph G' isolates C, so $\mathrm{dsep}_G(C, P, A)$ holds, whereas a search climbing from C to its parent P would report the opposite. Each test is $O(1)$.
Step 3, undirected connectivity. Breadth-first search on G' ignoring edge directions, seeded with all undeleted nodes of $\mathbf{X}$: pop V, report that $\mathrm{dsep}_G(\mathbf{X}, \mathbf{Z}, \mathbf{Y})$ fails if inY$[V]$, otherwise enqueue every unvisited undeleted neighbour reachable under the Step 2 rule; report that it holds if the queue empties first. Each node is visited once and each adjacency list scanned once, so this is $O(n+e)$ time and $O(n)$ space.
Steps 1 and 2 realise exactly the two pruning rules of Theorem 4.1, and no node of $\mathbf{X} \cup \mathbf{Y} \cup \mathbf{Z}$ is ever deleted, so the search reaches $\mathbf{Y}$ iff $\mathbf{X}$ and $\mathbf{Y}$ are connected in G', which by Theorem 4.1 is iff d-separation fails.

3.2 Exercises 4.8–4.14

Problem 4.8. Show that the graphoid axioms imply the chain rule

$$I(\mathbf{X}, \mathbf{Y}, \mathbf{Z}) \text{ and } I(\mathbf{X} \cup \mathbf{Y}, \mathbf{Z}, \mathbf{W}) \text{ only if } I(\mathbf{X}, \mathbf{Y}, \mathbf{W}),$$

where $\mathbf{X}, \mathbf{Y}, \mathbf{Z}, \mathbf{W}$ are pairwise disjoint sets of variables and $I(\mathbf{A}, \mathbf{B}, \mathbf{C})$ asserts that $\mathbf{A}$ and $\mathbf{C}$ are independent given $\mathbf{B}$.

Solution. Symmetry (4.3), weak union (4.7), contraction (4.9) and decomposition (4.4), in that order. Write (1) for $I(\mathbf{X}, \mathbf{Y}, \mathbf{Z})$ and (2) for $I(\mathbf{X} \cup \mathbf{Y}, \mathbf{Z}, \mathbf{W})$.
Symmetry turns (2) into $I(\mathbf{W}, \mathbf{Z}, \mathbf{X} \cup \mathbf{Y})$; weak union, $I(\mathbf{A}, \mathbf{S}, \mathbf{B} \cup \mathbf{C}) \Rightarrow I(\mathbf{A}, \mathbf{S} \cup \mathbf{B}, \mathbf{C})$ with $\mathbf{B} = \mathbf{Y}$, $\mathbf{C} = \mathbf{X}$, gives $I(\mathbf{W}, \mathbf{Z} \cup \mathbf{Y}, \mathbf{X})$; and symmetry again

$$(3) \quad I(\mathbf{X}, \mathbf{Y} \cup \mathbf{Z}, \mathbf{W}).$$

Contraction, $I(\mathbf{A}, \mathbf{S}, \mathbf{B})$ and $I(\mathbf{A}, \mathbf{S} \cup \mathbf{B}, \mathbf{C})$ only if $I(\mathbf{A}, \mathbf{S}, \mathbf{B} \cup \mathbf{C})$, has (1) and (3) as its two premises under $\mathbf{S} = \mathbf{Y}$, $\mathbf{B} = \mathbf{Z}$, $\mathbf{C} = \mathbf{W}$, so it yields $I(\mathbf{X}, \mathbf{Y}, \mathbf{Z} \cup \mathbf{W})$; decomposition then drops $\mathbf{Z}$, leaving

$$I(\mathbf{X}, \mathbf{Y}, \mathbf{W}).$$

Every set operation is legitimate because the premises force $\mathbf{X}, \mathbf{Y}, \mathbf{Z}, \mathbf{W}$ to be pairwise disjoint, the three arguments of an independence statement being disjoint.

Problem 4.9. Prove that the graphoid axioms hold for probability distributions, and that the intersection axiom holds for strictly positive distributions. That is, for a distribution Pr and pairwise disjoint sets of variables $\mathbf{X}, \mathbf{Y}, \mathbf{Z}, \mathbf{W}$, prove

- triviality: $I_{\Pr}(\mathbf{X}, \mathbf{Z}, \emptyset)$;
- symmetry: $I_{\Pr}(\mathbf{X}, \mathbf{Z}, \mathbf{Y})$ if and only if $I_{\Pr}(\mathbf{Y}, \mathbf{Z}, \mathbf{X})$;
- decomposition: $I_{\Pr}(\mathbf{X}, \mathbf{Z}, \mathbf{Y} \cup \mathbf{W})$ only if $I_{\Pr}(\mathbf{X}, \mathbf{Z}, \mathbf{Y})$ and $I_{\Pr}(\mathbf{X}, \mathbf{Z}, \mathbf{W})$;
- weak union: $I_{\Pr}(\mathbf{X}, \mathbf{Z}, \mathbf{Y} \cup \mathbf{W})$ only if $I_{\Pr}(\mathbf{X}, \mathbf{Z} \cup \mathbf{Y}, \mathbf{W})$;
- contraction: $I_{\Pr}(\mathbf{X}, \mathbf{Z}, \mathbf{Y})$ and $I_{\Pr}(\mathbf{X}, \mathbf{Z} \cup \mathbf{Y}, \mathbf{W})$ only if $I_{\Pr}(\mathbf{X}, \mathbf{Z}, \mathbf{Y} \cup \mathbf{W})$;
- intersection: $I_{\Pr}(\mathbf{X}, \mathbf{Z} \cup \mathbf{W}, \mathbf{Y})$ and $I_{\Pr}(\mathbf{X}, \mathbf{Z} \cup \mathbf{Y}, \mathbf{W})$ only if $I_{\Pr}(\mathbf{X}, \mathbf{Z}, \mathbf{Y} \cup \mathbf{W})$, whenever Pr is strictly positive.

Recall from Section 4.4 that $I_{\Pr}(\mathbf{X}, \mathbf{Z}, \mathbf{Y})$ means $\Pr(\mathbf{x} \mid \mathbf{z}, \mathbf{y}) = \Pr(\mathbf{x} \mid \mathbf{z})$ or $\Pr(\mathbf{y}, \mathbf{z}) = 0$, for all instantiations $\mathbf{x}, \mathbf{y}, \mathbf{z}$.

Solution. Two reformulations of the definition do all the work; throughout, $\mathbf{x}, \mathbf{y}, \mathbf{z}, \mathbf{w}$ range over instantiations and Pr of a partial instantiation is the corresponding marginal.

Lemma A (product form). $I_{\Pr}(\mathbf{X}, \mathbf{Z}, \mathbf{Y})$ iff

$$\Pr(\mathbf{x}, \mathbf{y}, \mathbf{z}) \Pr(\mathbf{z}) = \Pr(\mathbf{x}, \mathbf{z}) \Pr(\mathbf{y}, \mathbf{z}) \quad \text{for all } \mathbf{x}, \mathbf{y}, \mathbf{z}. \tag{A}$$

Indeed, if $\Pr(\mathbf{y}, \mathbf{z}) = 0$ then $\Pr(\mathbf{x}, \mathbf{y}, \mathbf{z}) = 0$ too and both sides of (A) vanish; otherwise $\Pr(\mathbf{z}) > 0$ and (A) is the defining equation multiplied by $\Pr(\mathbf{y}, \mathbf{z}) \Pr(\mathbf{z})$, a reversible step.

Lemma B (factorization). $I_{\Pr}(\mathbf{X}, \mathbf{Z}, \mathbf{Y})$ iff there are nonnegative f, g with

$$\Pr(\mathbf{x}, \mathbf{y}, \mathbf{z}) = f(\mathbf{x}, \mathbf{z})\, g(\mathbf{y}, \mathbf{z}) \quad \text{for all } \mathbf{x}, \mathbf{y}, \mathbf{z}. \tag{B}$$

($\Rightarrow$) Take $f = \Pr(\mathbf{x}, \mathbf{z})$ and $g = \Pr(\mathbf{y}, \mathbf{z}) / \Pr(\mathbf{z})$ when $\Pr(\mathbf{z}) > 0$, $g = 0$ otherwise; then (A) gives $fg = \Pr(\mathbf{x}, \mathbf{y}, \mathbf{z})$ in the first case, and both sides vanish in the second. ($\Leftarrow$) With $F(\mathbf{z}) = \sum_{\mathbf{x}} f$ and $G(\mathbf{z}) = \sum_{\mathbf{y}} g$, summing (B) gives $\Pr(\mathbf{x}, \mathbf{z}) = fG$, $\Pr(\mathbf{y}, \mathbf{z}) = Fg$ and $\Pr(\mathbf{z}) = FG$, whence

$$\Pr(\mathbf{x}, \mathbf{z}) \Pr(\mathbf{y}, \mathbf{z}) = fgFG = \Pr(\mathbf{x}, \mathbf{y}, \mathbf{z}) \Pr(\mathbf{z}),$$

which is (A).

Triviality. For $\mathbf{Y} = \emptyset$ the single empty instantiation gives $\Pr(\mathbf{x}, \mathbf{z}) = \Pr(\mathbf{x}, \mathbf{z}) \cdot 1$, which is (B) with $g \equiv 1$.

Symmetry. Condition (A) is unchanged when $\mathbf{X}$ and $\mathbf{Y}$ are interchanged.

Decomposition. By Lemma B the premise is $\Pr(\mathbf{x}, \mathbf{y}, \mathbf{w}, \mathbf{z}) = f(\mathbf{x}, \mathbf{z}) g(\mathbf{y}, \mathbf{w}, \mathbf{z})$; summing over $\mathbf{w}$ leaves $f(\mathbf{x}, \mathbf{z})\, g'(\mathbf{y}, \mathbf{z})$, again of the form (B), so $I_{\Pr}(\mathbf{X}, \mathbf{Z}, \mathbf{Y})$; summing over $\mathbf{y}$ instead gives $I_{\Pr}(\mathbf{X}, \mathbf{Z}, \mathbf{W})$.

Weak union. In that same factorization read $\mathbf{zy}$ as one instantiation of $\mathbf{Z} \cup \mathbf{Y}$ and set $f'(\mathbf{x}, \mathbf{zy}) = f(\mathbf{x}, \mathbf{z})$, $g'(\mathbf{w}, \mathbf{zy}) = g(\mathbf{y}, \mathbf{w}, \mathbf{z})$: this is (B) for the triple $(\mathbf{X}, \mathbf{Z} \cup \mathbf{Y}, \mathbf{W})$.

Contraction. If $\Pr(\mathbf{y}, \mathbf{w}, \mathbf{z}) = 0$ the required disjunct holds; otherwise $\Pr(\mathbf{y}, \mathbf{z}) > 0$ as well, and the second premise (at $\mathbf{w}$ and $\mathbf{zy}$) followed by the first gives

$$\Pr(\mathbf{x} \mid \mathbf{z}, \mathbf{y}, \mathbf{w}) = \Pr(\mathbf{x} \mid \mathbf{z}, \mathbf{y}) = \Pr(\mathbf{x} \mid \mathbf{z}).$$

Intersection, Pr strictly positive. Every instantiation now has nonzero probability, so all conditionals below are defined and the two premises read $\Pr(\mathbf{x} \mid \mathbf{z}, \mathbf{w}, \mathbf{y}) = \Pr(\mathbf{x} \mid \mathbf{z}, \mathbf{w})$ and $\Pr(\mathbf{x} \mid \mathbf{z}, \mathbf{y}, \mathbf{w}) = \Pr(\mathbf{x} \mid \mathbf{z}, \mathbf{y})$; their left-hand sides coincide, so

$$\Pr(\mathbf{x} \mid \mathbf{z}, \mathbf{w}) = \Pr(\mathbf{x} \mid \mathbf{z}, \mathbf{y}) \quad \text{for all } \mathbf{y}, \mathbf{w}. \tag{C}$$

Fixing one $\mathbf{w}_0$, (C) makes $\Pr(\mathbf{x} \mid \mathbf{z}, \mathbf{y}) = \Pr(\mathbf{x} \mid \mathbf{z}, \mathbf{w}_0) =: h(\mathbf{x}, \mathbf{z})$ for every $\mathbf{y}$, so

$$\Pr(\mathbf{x} \mid \mathbf{z}) = \sum_{\mathbf{y}} \Pr(\mathbf{x} \mid \mathbf{z}, \mathbf{y}) \Pr(\mathbf{y} \mid \mathbf{z}) = h(\mathbf{x}, \mathbf{z}),$$

and therefore $\Pr(\mathbf{x} \mid \mathbf{z}, \mathbf{y}, \mathbf{w}) = h(\mathbf{x}, \mathbf{z}) = \Pr(\mathbf{x} \mid \mathbf{z})$. Strict positivity is what makes (C) available for all pairs $\mathbf{y}, \mathbf{w}$ simultaneously.

Problem 4.10. Provide a probability distribution over three variables X, Y, and Z that violates the composition axiom. That is, show that $I_{\Pr}(Z, \emptyset, X)$ and $I_{\Pr}(Z, \emptyset, Y)$ but not $I_{\Pr}(Z, \emptyset, XY)$. Hint: Assume that X and Y are inputs to a noisy gate and Z is its output.

Solution. Take X and Y to be independent uniform binary inputs to an exclusive-or gate with output Z: the DAG $X \to Z \leftarrow Y$ with CPTs

$$\theta_{X=1} = \theta_{Y=1} = \tfrac{1}{2}, \qquad \theta_{Z=1|x,y} = [\, x \oplus y = 1 \,],$$

where $[\cdot]$ is 1 when the condition holds and 0 otherwise. The chain rule (Equation 4.2) induces

X	Y	Z	$\Pr(x, y, z)$
0	0	0	1/4
0	0	1	0
0	1	0	0
0	1	1	1/4
1	0	0	0
1	0	1	1/4
1	1	0	1/4
1	1	1	0

Summing the table, $\Pr(X = 1) = \Pr(Y = 1) = \Pr(Z = 1) = \frac{1}{2}$. Conditioning on $X = 1$ leaves Y uniform with $Z = 1$ exactly when $Y = 0$, and conditioning on $X = 0$ gives $Z = Y$, so

$$\Pr(Z = 1 \mid X = 1) = \Pr(Z = 1 \mid X = 0) = \tfrac{1}{2} = \Pr(Z = 1),$$

that is, $I_{\Pr}(Z, \emptyset, X)$; and $I_{\Pr}(Z, \emptyset, Y)$ by the symmetry of the gate and of the input distribution in X and Y. But the two inputs together determine the output:

$$\Pr(Z = 1 \mid X = 1, Y = 1) = 0 \neq \tfrac{1}{2} = \Pr(Z = 1),$$

so $I_{\Pr}(Z, \emptyset, XY)$ fails and composition (Equation 4.5) is violated.

Problem 4.11. Provide a probability distribution over three variables X, Y, and Z that violates the intersection axiom. That is, show that $I_{\Pr}(X, Z, Y)$ and $I_{\Pr}(X, Y, Z)$ but not $I_{\Pr}(X, \emptyset, YZ)$.

Solution. Take X, Y, Z binary and always equal, the common value set by a fair coin:

$$\Pr(X = Y = Z = 1) = \Pr(X = Y = Z = 0) = \tfrac{1}{2},$$

every other instantiation having probability 0. By Exercise 4.9 intersection holds for every strictly positive distribution, so a counterexample must encode a logical constraint, and equality is the simplest one.

X	Y	Z	$\Pr(x, y, z)$
0	0	0	1/2
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1/2

It is induced by $X \to Y \to Z$ with $\theta_{X=1} = \frac{1}{2}$ and the deterministic CPTs $\theta_{Y=y|X=x} = [\, y = x \,]$, $\theta_{Z=z|Y=y} = [\, z = y \,]$.
(i) $I_{\Pr}(X, Z, Y)$. Given $Z = z$, which has probability $\frac{1}{2} > 0$, all mass sits on $X = Y = z$; so $\Pr(y, z) > 0$ forces $y = z$ and then $\Pr(x \mid z, y) = [\, x = z \,] = \Pr(x \mid z)$, while for $y \neq z$ we have $\Pr(y, z) = 0$ and the second disjunct of the definition applies.

(ii) $I_{\Pr}(X, Y, Z)$. The same argument with Y and Z interchanged: $Y = y$ pins $X = y$, so $\Pr(x \mid y, z) = [\, x = y \,] = \Pr(x \mid y)$ whenever $\Pr(y, z) > 0$.
(iii) $I_{\Pr}(X, \emptyset, YZ)$ fails, since $\Pr(Y = 1, Z = 1) = \frac{1}{2} \neq 0$ and

$$\Pr(X = 1 \mid Y = 1, Z = 1) = 1 \neq \tfrac{1}{2} = \Pr(X = 1),$$

so neither disjunct holds. Intersection, Equation 4.13, is therefore violated.

Problem 4.12. Construct two distinct DAGs over variables A, B, C, and D. Each DAG must have exactly four edges and the DAGs must agree on d-separation, that is, for all disjoint sets of nodes $\mathbf{X}, \mathbf{Z}, \mathbf{Y}$ we must have $\mathrm{dsep}_{G_1}(\mathbf{X}, \mathbf{Z}, \mathbf{Y})$ if and only if $\mathrm{dsep}_{G_2}(\mathbf{X}, \mathbf{Z}, \mathbf{Y})$.

Solution. Take the two DAGs

$$\begin{aligned} G_1: &\quad A \to B, \quad B \to C, \quad C \to D, \quad A \to D, \\ G_2: &\quad B \to A, \quad B \to C, \quad C \to D, \quad A \to D. \end{aligned}$$

Both have four edges, both are acyclic (topological orders A, B, C, D and B, A, C, D), and they differ in the orientation of the edge between A and B. By Definition 4.2, $\mathrm{dsep}(\mathbf{X}, \mathbf{Z}, \mathbf{Y})$ holds iff $\mathrm{dsep}(X, \mathbf{Z}, Y)$ for every $X \in \mathbf{X}$ and $Y \in \mathbf{Y}$, so only singleton pairs need checking.
(i) Adjacent pairs. Both DAGs have the skeleton $A - B - C - D - A$, and a single-edge path carries no valve and is never blocked, so $\{A, B\}$, $\{B, C\}$, $\{C, D\}$, $\{A, D\}$ are d-separated by no $\mathbf{Z}$ in either DAG.
(ii) The pair $\{A, C\}$, where $\mathbf{Z} \subseteq \{B, D\}$. The two paths are $A \to B \to C$ (sequential at B) and $A \to D \leftarrow C$ in G_1, and $A \leftarrow B \to C$ (divergent at B) with the same $A \to D \leftarrow C$ in G_2. Sequential and divergent valves close under the identical condition $B \in \mathbf{Z}$, and D is a leaf in both DAGs, so both give:

$\mathbf{Z}$	valve at B	valve at D	d-separated?
$\emptyset$	open	closed	no
$\{B\}$	closed	closed	yes
$\{D\}$	open	open	no
$\{B, D\}$	closed	open	no

So in both DAGs A and C are d-separated exactly by $\mathbf{Z} = \{B\}$.
(iii) The pair $\{B, D\}$, where $\mathbf{Z} \subseteq \{A, C\}$. The paths are $B \to C \to D$ together with $B \leftarrow A \to D$ in G_1 and $B \to A \to D$ in G_2; every valve is again sequential or divergent, hence closed exactly when its node lies in $\mathbf{Z}$:

$\mathbf{Z}$	valve at C	valve at A	d-separated?
$\emptyset$	open	open	no
$\{A\}$	open	closed	no
$\{C\}$	closed	open	no
$\{A, C\}$	closed	closed	yes

Again identical: B and D are d-separated exactly by $\mathbf{Z} = \{A, C\}$. Agreeing on every singleton query, G_1 and G_2 agree on every d-separation query.

Problem 4.13. Prove that d-separation satisfies the properties of intersection and chordality, stated in Section 4.5.3 as

$$\mathrm{dsep}(\mathbf{X}, \mathbf{Z} \cup \mathbf{W}, \mathbf{Y}) \text{ and } \mathrm{dsep}(\mathbf{X}, \mathbf{Z} \cup \mathbf{Y}, \mathbf{W}) \text{ only if } \mathrm{dsep}(\mathbf{X}, \mathbf{Z}, \mathbf{Y} \cup \mathbf{W})$$

for disjoint sets of nodes $\mathbf{X}, \mathbf{Y}, \mathbf{Z}, \mathbf{W}$, and

$$\mathrm{dsep}(X, \{Z, W\}, Y) \text{ and } \mathrm{dsep}(W, \{X, Y\}, Z) \text{ only if } \mathrm{dsep}(X, Z, Y) \text{ or } \mathrm{dsep}(X, W, Y)$$

for distinct single nodes X, Y, Z, W.

Solution. Both hold, with blocking as in Definition 4.2.
Observation. If $\mathbf{S} \subseteq \mathbf{S}'$ and a valve of a path is closed given $\mathbf{S}'$ but open given $\mathbf{S}$, it is sequential or divergent with node in $\mathbf{S}' \setminus \mathbf{S}$: such a valve is closed exactly when its node lies in the conditioning set, while a convergent valve closed given $\mathbf{S}'$ is a fortiori closed given the smaller $\mathbf{S}$, which contains neither its node nor a descendant.
Intersection. Assume (1) $\mathrm{dsep}(\mathbf{X}, \mathbf{Z} \cup \mathbf{W}, \mathbf{Y})$ and (2) $\mathrm{dsep}(\mathbf{X}, \mathbf{Z} \cup \mathbf{Y}, \mathbf{W})$, and suppose the conclusion fails; the premises force $\mathbf{X}, \mathbf{Y}, \mathbf{Z}, \mathbf{W}$ pairwise disjoint. Let π be a shortest path from $s \in \mathbf{X}$ to $t \in \mathbf{Y} \cup \mathbf{W}$ unblocked by $\mathbf{Z}$.

- (i) $t \in \mathbf{Y}$: by (1) some valve of π is closed given $\mathbf{Z} \cup \mathbf{W}$, and it is open given $\mathbf{Z}$, so by the Observation it is sequential or divergent at an interior node $V \in \mathbf{W}$.
- (ii) $t \in \mathbf{W}$: by (2), symmetrically, at an interior node $V \in \mathbf{Y}$.

Either way the sub-path of π from s to V carries the same valve types at the same interior nodes, so it too is unblocked by $\mathbf{Z}$; it runs from $\mathbf{X}$ into $\mathbf{Y} \cup \mathbf{W}$ and is shorter, contradicting the choice of π. Hence $\mathrm{dsep}(\mathbf{X}, \mathbf{Z}, \mathbf{Y} \cup \mathbf{W})$.
Chordality. Write $An(\mathbf{S})$ for $\mathbf{S}$ with all ancestors of its members and $M(\mathbf{S})$ for the moral graph of G restricted to $An(\mathbf{S})$. By Exercise 4.24, for disjoint sets $\mathrm{dsep}_G(\mathbf{X}, \mathbf{Z}, \mathbf{Y})$ iff $\mathbf{Z}$ separates $\mathbf{X}$ from $\mathbf{Y}$ in $M(\mathbf{X} \cup \mathbf{Y} \cup \mathbf{Z})$. Monotonicity: for $\mathbf{S}' \subseteq \mathbf{S}$, $M(\mathbf{S}')$ is a subgraph of $M(\mathbf{S})$, since $An(\mathbf{S}') \subseteq An(\mathbf{S})$ and a marriage through a common child $c \in An(\mathbf{S}')$ is a marriage through $c \in An(\mathbf{S})$. (Check!)
Assume (1) $\mathrm{dsep}(X, \{Z, W\}, Y)$ and (2) $\mathrm{dsep}(W, \{X, Y\}, Z)$ for distinct X, Y, Z, W, and for contradiction that (3) $\mathrm{dsep}(X, Z, Y)$ and (4) $\mathrm{dsep}(X, W, Y)$ both fail. Put $A = An(\{X, Y, Z, W\})$, $M = M(\{X, Y, Z, W\})$, $M_1 = M(\{X, Y, Z\})$, $M_2 = M(\{X, Y, W\})$; premises (1) and (2) are read in M, and M_1, M_2 are subgraphs of M by monotonicity.

1. X and Y are non-adjacent in M: an edge would be an X-Y path meeting neither Z nor W, against (1).
2. By (3) and Exercise 4.24 there is an X-Y path p of M_1, hence of M, avoiding Z; by 1 it has an interior node and by (1) it meets $\{Z, W\}$, so it contains W. Then $W \in An(\{X, Y, Z\}) \setminus \{X, Y, Z\}$, i.e. W is a proper ancestor of X, Y, or Z. Symmetrically (4) gives an X-Y path q of M avoiding W and containing Z, with Z a proper ancestor of X, Y, or W.
3. In $M^- = M$ minus X and Y, premise (2) puts W and Z in distinct components $C_W \neq C_Z$ joined by no edge of M, and $\mathrm{interior}(p) \subseteq C_W$, $\mathrm{interior}(q) \subseteq C_Z$.
4. Let R be the last of X, Y, Z, W in a topological order of G; R is a proper ancestor of none of the others, which by 2 excludes Z and W, so $R \in \{X, Y\}$. All of (1), (3), (4) are symmetric in their outer arguments, (2) mentions $\{X, Y\}$ as a set, and A, M, M_1, M_2 are unchanged by the swap, so take $R = Y$.
5. Y has no child in A: a child $U \in A$ has $U \notin An(Y)$ by acyclicity, hence is an ancestor of or equal to one of X, Z, W, making Y a proper ancestor of one of them against 4.
6. So every neighbour of Y in M is a parent of Y in $G[A]$: the other two sources of an M-edge at Y, namely Y being a parent of the neighbour or the two being co-parents of a node of A, both need Y to have a child in A.
7. Let c and d be the neighbours of Y on p and on q; by 1 and 3 they are interior, with $c \in C_W$ and $d \in C_Z$, so $c \neq d$. By 6 both are parents of Y, so moralization puts $c - d$ in M, joining C_W to C_Z against 3.

Hence (3) and (4) cannot both fail: $\mathrm{dsep}(X, Z, Y)$ or $\mathrm{dsep}(X, W, Y)$.

Problem 4.14. Consider the DAG G in Figure 4.4, over the five propositional variables A (Winter?), B (Sprinkler?), C (Rain?), D (Wet Grass?) and E (Slippery Road?), with edges

$$A \to B, \quad A \to C, \quad B \to D, \quad C \to D, \quad C \to E.$$

Suppose that this DAG is a P-MAP of some distribution Pr. Construct a minimal I-MAP G' for Pr using each of the following variable orders:

- (a) A, D, B, C, E
- (b) A, B, C, D, E
- (c) E, D, C, B, A

Solution. Run the procedure of Section 4.6.2, giving X_i as parents a minimal $\mathbf{P} \subseteq \{X_1, \ldots, X_{i-1}\}$ with $I_{\Pr}(X_i, \mathbf{P}, \{X_1, \ldots, X_{i-1}\} \setminus \mathbf{P})$; since G is a P-MAP of Pr, each test is the corresponding dsep_G test. Two facts shorten every step: a predecessor adjacent to X_i lies in every admissible $\mathbf{P}$ (a single edge carries no valve, Definition 4.2), and D is a leaf, so a convergent valve at D is closed exactly when $D \notin \mathbf{P}$.

(a) Order A, D, B, C, E.

- A: $\mathbf{P} = \emptyset$.
- D: $\mathrm{dsep}_G(D, \emptyset, A)$ fails, the sequential valve at B on $A \to B \to D$ being open, so $\mathbf{P} = \{A\}$.
- B: adjacent to A and D, so $\mathbf{P} = \{A, D\}$.
- C: adjacent to A and D, and $\mathrm{dsep}_G(C, \{A, D\}, B)$ fails because the convergent valve at D on $C \to D \leftarrow B$ is open; so $\mathbf{P} = \{A, B, D\}$.
- E: adjacent to C, and $\{C\}$ suffices, every path out of E beginning $E \leftarrow C$ and so carrying a sequential or divergent valve at C.

$$A \to D, \quad A \to B, \quad D \to B, \quad A \to C, \quad B \to C, \quad D \to C, \quad C \to E.$$

(b) Order A, B, C, D, E, a topological order of G, returns G itself:

- A: $\emptyset$; B: adjacent to A, so $\{A\}$.
- C: adjacent to A, and $\{A\}$ suffices, since $C \leftarrow A \to B$ is closed at the divergent valve A and $C \to D \leftarrow B$ at the convergent valve D.
- D: adjacent to B and C, and $\{B, C\}$ suffices, the paths $D \leftarrow B \leftarrow A$ and $D \leftarrow C \leftarrow A$ being closed at their sequential valves.
- E: $\{C\}$ as in (a).

$$A \to B, \quad A \to C, \quad B \to D, \quad C \to D, \quad C \to E.$$

(c) Order E, D, C, B, A.

- E: $\emptyset$.
- D: $\mathrm{dsep}_G(D, \emptyset, E)$ fails at the divergent valve C of $D \leftarrow C \to E$, so $\mathbf{P} = \{E\}$.
- C: adjacent to D and E, so $\mathbf{P} = \{D, E\}$.
- B: adjacent to D, and $\{C, D\}$ passes, since $B \leftarrow A \to C \to E$ is closed at the sequential valve C and $B \to D \leftarrow C \to E$ at the divergent valve C (its valve at D being open). Both $\{D\}$ and $\{D, E\}$ leave C on the right, where $B \leftarrow A \to C$ is unblocked. So $\mathbf{P} = \{C, D\}$.
- A: adjacent to B and C, and $\{B, C\}$ suffices: $A \to B \to D$, $A \to C \to D$ and $A \to C \to E$ are blocked at their sequential valves, and $A \to B \to D \leftarrow C \to E$ at B.

$$E \to D, \quad E \to C, \quad D \to C, \quad C \to B, \quad D \to B, \quad B \to A, \quad C \to A.$$

3.3 Exercises 4.15–4.21

Problem 4.15. Identify a DAG that is a D-MAP for all distributions Pr over variables $\mathbf{X}$. Similarly, identify another DAG that is an I-MAP for all distributions Pr over variables $\mathbf{X}$.

Solution. The edgeless DAG $G_\emptyset$ over $\mathbf{X}$ is a D-MAP of every distribution, and the complete DAG G_c on any fixed ordering $X_1, \ldots, X_n$ (edges $X_i \to X_j$ for $i < j$) is an I-MAP of every distribution. Both conditions of Section 4.6 are implications, and each graph satisfies its own vacuously.

D-MAP. $G_\emptyset$ has no path at all between a node of $\mathbf{X}_1$ and a node of $\mathbf{Y}$, so the blocking condition of Definition 4.2 is met vacuously and $\mathrm{dsep}_{G_\emptyset}(\mathbf{X}_1, \mathbf{Z}, \mathbf{Y})$ holds for every disjoint triple. The consequent of

$$I_{\Pr}(\mathbf{X}_1, \mathbf{Z}, \mathbf{Y}) \text{ only if } \mathrm{dsep}_{G_\emptyset}(\mathbf{X}_1, \mathbf{Z}, \mathbf{Y})$$

is thus always true, for every Pr.

I-MAP. G_c is acyclic, all edges running forward in the ordering. If $\mathbf{X}_1, \mathbf{Z}, \mathbf{Y}$ are disjoint with $\mathbf{X}_1$ and $\mathbf{Y}$ nonempty, pick $X \in \mathbf{X}_1$ and $Y \in \mathbf{Y}$; the edge joining them is a path with no valves, hence never

blocked, so $\mathrm{dsep}_{G_c}(\mathbf{X}_1, \mathbf{Z}, \mathbf{Y})$ fails. D-separation holds in G_c only when $\mathbf{X}_1 = \emptyset$ or $\mathbf{Y} = \emptyset$, and for those triples I_{Pr} holds by triviality. So the antecedent of

$$\mathrm{dsep}_{G_c}(\mathbf{X}_1, \mathbf{Z}, \mathbf{Y}) \text{ only if } I_{\mathrm{Pr}}(\mathbf{X}_1, \mathbf{Z}, \mathbf{Y})$$

is never nontrivially true, for every Pr.

Problem 4.16. Consider the DAG G in Figure 4.15. This DAG has nine nodes, A_1, A_2, A_3, B_1, B_2, B_3, C_1, C_2, C_3, arranged in three rows of three, and the following ten edges:

$$\begin{array}{llll} A_1 \to A_2, & A_2 \to A_3, & A_1 \to B_2, & A_2 \to B_3, \\ B_1 \to B_2, & B_2 \to B_3, & B_1 \to C_2, & B_2 \to C_3, \\ C_1 \to C_2, & C_2 \to C_3. & & \end{array}$$

That is, each node in row A points to the next node in its own row and to the next node in row B; each node in row B points to the next node in its own row and to the next node in row C; each node in row C points only to the next node in row C. Suppose that this DAG is a P-MAP of a distribution Pr.
(a) What is the Markov boundary for the variable C_2?
(b) Is the Markov boundary of A_1 a Markov blanket of B_3?
(c) Which variable has the smallest Markov boundary?

Solution. (a) $\{B_1, C_1, C_3, B_2\}$; (b) yes; (c) A_3, whose boundary $\{A_2\}$ has size one.
By Corollary 1 the set $\mathrm{mb}(X)$ of parents, children, and spouses of X (a spouse being another parent of a child of X) is a Markov blanket for X; under the P-MAP hypothesis it is the Markov boundary, since it sits inside every blanket $\mathbf{B}$. For $I_{\mathrm{Pr}}(X, \mathbf{B}, \mathbf{X} \setminus \mathbf{B} \setminus \{X\})$ gives $\mathrm{dsep}_G(X, \mathbf{B}, \mathbf{X} \setminus \mathbf{B} \setminus \{X\})$, G being a D-MAP, and any $Y \in \mathrm{mb}(X) \setminus \mathbf{B}$ refutes that d-separation:

- Y a parent or child of X: the edge between them is a valveless path, never blocked;
- Y a spouse through a child $Z \in \mathbf{B}$: the only valve of $X \to Z \leftarrow Y$ is convergent at Z, hence open;
- Y a spouse through a child $Z \notin \mathbf{B}$: the edge $X \to Z$ is itself an unblocked path into the third argument.

(a) C_2 has parents B_1, C_1, sole child C_3, and through C_3 the single spouse B_2:

$$\mathrm{mb}(C_2) = \{B_1, C_1, C_3, B_2\}.$$

(b) Yes. The root A_1 has children A_2, B_2 and one spouse B_1 (the other parent of B_2), so $\mathrm{mb}(A_1) = \{A_2, B_2, B_1\}$; the leaf B_3 has parents A_2, B_2, so $\mathrm{mb}(B_3) = \{A_2, B_2\}$ is a blanket of B_3 by Corollary 1. Since $\{A_2, B_2\} \subseteq \mathrm{mb}(A_1)$ and $B_3 \notin \mathrm{mb}(A_1)$, it is enough that a superset $\mathbf{B} \supseteq \mathbf{S}$ of a blanket with $X \notin \mathbf{B}$ is again a blanket: splitting $\mathbf{X} \setminus \mathbf{S} \setminus \{X\}$ as $(\mathbf{B} \setminus \mathbf{S}) \cup (\mathbf{X} \setminus \mathbf{B} \setminus \{X\})$, weak union turns $I_{\mathrm{Pr}}(X, \mathbf{S}, \mathbf{X} \setminus \mathbf{S} \setminus \{X\})$ into

$$I_{\mathrm{Pr}}\big(X,\ \mathbf{B},\ \mathbf{X} \setminus \mathbf{B} \setminus \{X\}\big).$$

(c) Tabulating parents, children, and spouses for all nine nodes:

node	parents	children	spouses	boundary	size
A_1	—	A_2, B_2	B_1	$\{A_2, B_2, B_1\}$	3
A_2	A_1	A_3, B_3	B_2	$\{A_1, A_3, B_3, B_2\}$	4
A_3	A_2	—	—	$\{A_2\}$	1
B_1	—	B_2, C_2	A_1, C_1	$\{B_2, C_2, A_1, C_1\}$	4
B_2	A_1, B_1	B_3, C_3	A_2, C_2	$\{A_1, B_1, B_3, C_3, A_2, C_2\}$	6
B_3	A_2, B_2	—	—	$\{A_2, B_2\}$	2
C_1	—	C_2	B_1	$\{C_2, B_1\}$	2
C_2	B_1, C_1	C_3	B_2	$\{B_1, C_1, C_3, B_2\}$	4
C_3	B_2, C_2	—	—	$\{B_2, C_2\}$	2

The smallest is $\mathrm{mb}(A_3) = \{A_2\}$, and it is a boundary: its only proper subset $\emptyset$ fails, since the edge

$A_2 \to A_3$ is an unblockable path and the D-MAP property then denies A_3 marginal independence of the rest.

Problem 4.17. Prove that for strictly positive distributions, if $\mathbf{B}_1$ and $\mathbf{B}_2$ are Markov blankets for some variable X, then $\mathbf{B}_1 \cap \mathbf{B}_2$ is also a Markov blanket for X. Hint: Appeal to the intersection axiom.

Solution. Partition $\mathbf{R} = \mathbf{X} \setminus \{X\}$ into

$$\mathbf{S} = \mathbf{B}_1 \cap \mathbf{B}_2, \qquad \mathbf{D}_1 = \mathbf{B}_1 \setminus \mathbf{B}_2,$$
$$\mathbf{D}_2 = \mathbf{B}_2 \setminus \mathbf{B}_1, \qquad \mathbf{O} = \mathbf{R} \setminus (\mathbf{B}_1 \cup \mathbf{B}_2),$$

so that by Definition 4.3 the hypotheses read $I_{\Pr}(X, \mathbf{S} \cup \mathbf{D}_1, \mathbf{D}_2 \cup \mathbf{O})$ and $I_{\Pr}(X, \mathbf{S} \cup \mathbf{D}_2, \mathbf{D}_1 \cup \mathbf{O})$, while the goal is $I_{\Pr}(X, \mathbf{S}, \mathbf{D}_1 \cup \mathbf{D}_2 \cup \mathbf{O})$; the side condition $X \notin \mathbf{S}$ follows from $X \notin \mathbf{B}_1$. Weak union (4.7) and the intersection axiom (4.13), the latter available because Pr is strictly positive, give

$$\begin{aligned}
&(1) \quad I_{\Pr}(X,\ \mathbf{S} \cup \mathbf{D}_1 \cup \mathbf{O},\ \mathbf{D}_2),\\
&(2) \quad I_{\Pr}(X,\ \mathbf{S} \cup \mathbf{D}_2 \cup \mathbf{O},\ \mathbf{D}_1),\\
&(3) \quad I_{\Pr}(X,\ \mathbf{S} \cup \mathbf{O},\ \mathbf{D}_1 \cup \mathbf{D}_2),\\
&(4) \quad I_{\Pr}(X,\ \mathbf{S} \cup \mathbf{D}_1 \cup \mathbf{D}_2,\ \mathbf{O}),
\end{aligned}$$

where (1) and (2) are weak union on the two hypotheses with $\mathbf{Y} = \mathbf{O}$; (3) is intersection on (1) and (2) with $\mathbf{Z} = \mathbf{S} \cup \mathbf{O}$, $\mathbf{W} = \mathbf{D}_1$, $\mathbf{Y} = \mathbf{D}_2$; and (4) is weak union on the first hypothesis with $\mathbf{Y} = \mathbf{D}_2$, $\mathbf{W} = \mathbf{O}$. Intersection once more, on (3) and (4) with $\mathbf{Z} = \mathbf{S}$, $\mathbf{W} = \mathbf{O}$, $\mathbf{Y} = \mathbf{D}_1 \cup \mathbf{D}_2$, yields

$$I_{\Pr}(X,\ \mathbf{S},\ \mathbf{D}_1 \cup \mathbf{D}_2 \cup \mathbf{O}),$$

which with $X \notin \mathbf{S}$ is exactly the statement that $\mathbf{B}_1 \cap \mathbf{B}_2$ is a Markov blanket for X. ■

Problem 4.18. (After Pearl) Consider the following independence statements over the four variables A, B, C, D: $I(A, \emptyset, B)$ and $I(AB, C, D)$.
(a) Find all independence statements that follow from these two statements using the positive graphoid axioms.
(b) Construct minimal I-MAPs of the statements in (a) (original and derived) using the following variable orders:

- A, B, C, D
- D, C, B, A
- A, D, B, C

Solution. (a) The closure is the twelve statements displayed below; (b) the three orders return G_1 (three edges), a four-edge DAG, and the complete DAG.
(a) From $I(A, \emptyset, B)$ only symmetry is productive, both outer arguments being singletons and the conditioning set empty. From $I(AB, C, D)$, symmetry gives $I(D, C, AB)$, whose right argument is a pair: decomposition then gives $I(D, C, A)$ and $I(D, C, B)$, weak union (4.7) gives $I(D, \{C, B\}, A)$ and $I(D, \{C, A\}, B)$, and symmetry doubles each. Contraction (4.9) on $I(D, C, A)$ and $I(D, \{C, A\}, B)$ returns $I(D, C, AB)$, as does intersection (4.13) on $I(D, \{C, A\}, B)$ and $I(D, \{C, B\}, A)$, so the derivation has stabilized; modulo the triviality statements $I(\mathbf{X}_1, \mathbf{Z}, \emptyset)$:

$$
\begin{array}{ll}
I(A,\emptyset,B), & I(B,\emptyset,A),\\
I(AB,C,D), & I(D,C,AB),\\
I(A,C,D), & I(D,C,A),\\
I(B,C,D), & I(D,C,B),\\
I(A,\{B,C\},D), & I(D,\{B,C\},A),\\
I(B,\{A,C\},D), & I(D,\{A,C\},B).
\end{array}
$$

That the list is closed is proved rather than inspected. Let G_1 be the DAG $A \to C$, $B \to C$, $C \to D$; its d-separations over disjoint triples are exactly the twelve statements above, both seeds among them, since A and B are non-adjacent roots joined only through the convergent valve at C and C blocks every path from $\{A, B\}$ to D. D-separation obeys every positive graphoid axiom — symmetry, decomposition, weak union, contraction and triviality from Definition 4.2, intersection from Exercise 4.13 — so the set of triples d-separated in G_1 is closed under the axioms and contains the seeds, hence contains the closure, hence equals it. Note that C never occurs as an outer argument.

(b) By Section 4.6.2, $\mathbf{P} \subseteq \{X_1, \ldots, X_{i-1}\}$ is admissible for X_i exactly when $I(X_i, \mathbf{P}, \{X_1, \ldots, X_{i-1}\} \setminus \mathbf{P})$ lies in the closure, and with Pr strictly positive the minimal such $\mathbf{P}$ is unique.

Order A, B, C, D, giving G_1 itself, hence a P-MAP of the statements in (a):

- A: $\emptyset$. B: $\emptyset$, by $I(B, \emptyset, A)$.
- C: none of $I(C, \emptyset, AB)$, $I(C, A, B)$, $I(C, B, A)$ is in the closure, so $\{A, B\}$.
- D: $I(D, C, AB)$ is in the closure and $\emptyset$, $\{A\}$, $\{B\}$ fail, so $\{C\}$.

Order D, C, B, A, giving $D \to C$, $C \to B$, $C \to A$, $B \to A$:

- D: $\emptyset$. C: $I(C, \emptyset, D)$ absent, so $\{D\}$.
- B: $I(B, \emptyset, \{C, D\})$ and $I(B, D, C)$ absent, $I(B, C, D)$ present, so $\{C\}$.
- A: $I(A, \emptyset, \{B, C, D\})$, $I(A, B, \{C, D\})$, $I(A, C, \{B, D\})$, $I(A, D, \{B, C\})$ all absent, $I(A, \{B, C\}, D)$ present, so $\{B, C\}$.

This is an I-MAP, the d-separations it asserts being $\mathrm{dsep}(D, C, A)$, $\mathrm{dsep}(D, C, B)$, $\mathrm{dsep}(D, C, AB)$, $\mathrm{dsep}(D, \{B, C\}, A)$ and $\mathrm{dsep}(D, \{A, C\}, B)$, all in the closure; it is not a P-MAP, the edge $B \to A$ destroying $I(A, \emptyset, B)$.

Order A, D, B, C, giving the complete DAG $A \to D$, $A \to B$, $A \to C$, $D \to B$, $D \to C$, $B \to C$:

- A: $\emptyset$. D: $I(D, \emptyset, A)$ absent, only $I(D, C, A)$ being present, so $\{A\}$.
- B: $I(B, \emptyset, \{A, D\})$ is absent, since composition is not a graphoid axiom and so $I(B, \emptyset, A)$ with $I(B, C, D)$ does not combine; $I(B, A, D)$ and $I(B, D, A)$ are absent too, so $\{A, D\}$.
- C: no closure statement has C as an outer argument, so $\{A, D, B\}$.

Asserting no independence at all, it is trivially an I-MAP, and minimal by Exercise 4.19: deleting $A \to C$, for instance, would assert $\mathrm{dsep}(C, \{D, B\}, A)$, absent from the closure.

Problem 4.19. Assume that the algorithm in Section 4.6.2 is correct as far as producing an I-MAP G for the given distribution Pr. Prove that G must also be a minimal I-MAP.

Recall the algorithm: given an ordering $X_1, \ldots, X_n$ of the variables of Pr, start with an edgeless DAG and process the variables in order; for each X_i identify a minimal subset $\mathbf{P}_i \subseteq \{X_1, \ldots, X_{i-1}\}$ such that $I_{\Pr}(X_i, \mathbf{P}_i, \{X_1, \ldots, X_{i-1}\} \setminus \mathbf{P}_i)$, and make $\mathbf{P}_i$ the parents of X_i.

Solution. Delete an arbitrary edge $U \to X_i$ of G, obtaining G'; we exhibit a d-separation that holds in G' while the matching independence fails in Pr, so that G' is not an I-MAP. Every edge of G issues from a member of $\mathbf{P}_i$ into X_i, so $U \in \mathbf{P}_i$ and in G' the parents of X_i are $\mathbf{P}'_i = \mathbf{P}_i \setminus \{U\}$, those of every other node unchanged. Put

$$
\mathbf{V}_i = \{X_1, \ldots, X_{i-1}\}, \qquad \mathbf{Y} = \mathbf{V}_i \setminus \mathbf{P}'_i.
$$

Claim: $\mathrm{dsep}_{G'}(X_i, \mathbf{P}'_i, \mathbf{Y})$. Apply Theorem 4.1 with $\mathbf{X}_1 \cup \mathbf{Z} \cup \mathbf{Y} = \{X_1, \ldots, X_i\}$. The algorithm draws edges only from earlier to later variables, so $X_1, \ldots, X_n$ is a topological order of G' and deleting $X_n, X_{n-1}, \ldots, X_{i+1}$ in turn prunes leaves lying outside $\{X_1, \ldots, X_i\}$. Deleting next the edges outgoing from $\mathbf{Z} = \mathbf{P}'_i$ isolates X_i: its outgoing edges went with its pruned children, and its incoming edges all

issue from $\mathbf{P}'_i$. Theorem 4.1 then gives the claim, which is just the local Markov property, $\mathbf{Y}$ consisting of non-descendants of X_i.
Were G' an I-MAP of Pr, the claim would force

$$I_{\Pr}\big(X_i,\ \mathbf{P}_i \setminus \{U\},\ \{X_1, \ldots, X_{i-1}\} \setminus (\mathbf{P}_i \setminus \{U\})\big),$$

exhibiting $\mathbf{P}_i \setminus \{U\}$ as a proper subset of $\mathbf{P}_i$ meeting the very condition the algorithm minimized over, a contradiction. As the edge was arbitrary, G is a minimal I-MAP of Pr. ■

Problem 4.20. Suppose that G is a DAG and let $\mathbf{W}$ be a set of nodes in G with deterministic CPTs (i.e., their parameters are either 0 or 1). Propose a modification to the d-separation test that can take advantage of nodes $\mathbf{W}$ and that will be stronger than d-separation (i.e., discover independencies that d-separation cannot discover).

Solution. Run the ordinary test on the **determined set** $\det(\mathbf{Z})$ in place of $\mathbf{Z}$: for disjoint $\mathbf{X}, \mathbf{Z}, \mathbf{Y}$, declare $\mathbf{X}$ and $\mathbf{Y}$ independent given $\mathbf{Z}$ whenever

$$\mathrm{dsep}_G\big(\mathbf{X} \setminus \det(\mathbf{Z}),\ \det(\mathbf{Z}),\ \mathbf{Y} \setminus \det(\mathbf{Z})\big),$$

where $\det(\mathbf{Z})$ is the least set with $\mathbf{Z} \subseteq \det(\mathbf{Z})$ and $V \in \det(\mathbf{Z})$ whenever $V \in \mathbf{W}$ and $\mathrm{Parents}(V) \subseteq \det(\mathbf{Z})$. The point is that a node of $\mathbf{W}$ is a function of its parents, exactly one of its parameters being 1 per parent instantiation, so knowledge propagates beyond the evidence set, which the ordinary test cannot see. One topological sweep computes the closure, so the test stays linear-time.
Soundness. By Exercise 4.5(c), $\Pr(v \mid \mathbf{u}) = \theta_{v|\mathbf{u}}$, so each $V \in \mathbf{W}$ with $\mathbf{U} = \mathrm{Parents}(V)$ has a unique value $v = f_V(\mathbf{u})$ of parameter 1, whence $\Pr(V \neq f_V(\mathbf{u}), \mathbf{u}) = 0$ and $\Pr(f_V(\mathbf{u}) \mid e) = 1$ for every positive-probability e implying $\mathbf{U} = \mathbf{u}$. Enumerate $\mathbf{D} = \det(\mathbf{Z}) \setminus \mathbf{Z}$ as $V_1, \ldots, V_k$ in the order the closure added them, so that $\mathrm{Parents}(V_j) \subseteq \mathbf{Z} \cup \{V_1, \ldots, V_{j-1}\}$, and fix $\mathbf{z}$ with $\Pr(\mathbf{z}) > 0$. If $\Pr(v_1 \cdots v_{j-1} \mid \mathbf{z}) = 1$ then $\mathbf{z}v_1 \cdots v_{j-1}$ is a positive-probability event fixing $\mathrm{Parents}(V_j)$, so $\Pr(v_j \mid \mathbf{z}v_1 \cdots v_{j-1}) = 1$ and the product is again 1; at $j = k$ a unique $\mathbf{d}$ has $\Pr(\mathbf{d} \mid \mathbf{z}) = 1$, and hence $\Pr(\alpha \mid \mathbf{z}) = \Pr(\alpha \mid \mathbf{z}, \mathbf{d})$ for every event α. Split $\mathbf{x}$ into its part $\mathbf{x}_1$ over $\mathbf{X} \setminus \det(\mathbf{Z})$ and $\mathbf{x}_2$ over $\mathbf{X} \cap \det(\mathbf{Z})$, and likewise $\mathbf{y}$. If $\mathbf{x}_2$ or $\mathbf{y}_2$ contradicts $\mathbf{zd}$, both sides of the independence equation vanish; otherwise $\mathbf{zd}$ implies them and

$$\begin{aligned}\Pr(\mathbf{x}, \mathbf{y} \mid \mathbf{z}) &= \Pr(\mathbf{x}_1, \mathbf{y}_1 \mid \mathbf{zd}) \\ &= \Pr(\mathbf{x}_1 \mid \mathbf{zd})\ \Pr(\mathbf{y}_1 \mid \mathbf{zd}) = \Pr(\mathbf{x} \mid \mathbf{z})\ \Pr(\mathbf{y} \mid \mathbf{z}),\end{aligned}$$

the middle equality being Theorem 4.2 applied to the displayed d-separation, since $\mathbf{zd}$ is an instantiation of $\det(\mathbf{Z})$, and the last holding because $\mathbf{zd}$ implies $\mathbf{x}_2$ and $\mathbf{y}_2$. Hence $I_{\Pr}(\mathbf{X}, \mathbf{Z}, \mathbf{Y})$.
Never weaker. Let $\mathrm{dsep}_G(\mathbf{X}, \mathbf{Z}, \mathbf{Y})$ hold and let p run from $\mathbf{X} \setminus \det(\mathbf{Z})$ to $\mathbf{Y} \setminus \det(\mathbf{Z})$; being also an $\mathbf{X}$-$\mathbf{Y}$ path, it has a valve closed given $\mathbf{Z}$.

- (i) Sequential or divergent at $N \in \mathbf{Z}$: then $N \in \det(\mathbf{Z})$ and the valve is still closed.
- (ii) Convergent at N, with no descendant of N in $\mathbf{Z}$. If $N \notin \det(\mathbf{Z})$ then no descendant M of N is in $\det(\mathbf{Z})$ either: every node on a directed path from N to such an M is a descendant of N, hence outside $\mathbf{Z}$, hence in $\det(\mathbf{Z}) \setminus \mathbf{Z}$, which drags its parents in and would force $N \in \det(\mathbf{Z})$. So the valve stays closed. If instead $N \in \det(\mathbf{Z}) \setminus \mathbf{Z}$, all parents of N lie in $\det(\mathbf{Z})$, in particular the two neighbours of N on p; each has an edge into N, so it is not convergent, and it is either an endpoint removed from the modified query or a sequential or divergent valve in $\det(\mathbf{Z})$, closed.

Either way p is blocked, so the modified test succeeds whenever the original does.
Strictly stronger. Take $A \to V \leftarrow B$, $V \to C$, $V \to D$ with binary variables, $\mathbf{W} = \{V\}$ and the deterministic CPT $V = A \oplus B$, and ask whether C and D are independent given $\{A, B\}$. Plain d-separation says no, the divergent valve V on $C \leftarrow V \to D$ being open since $V \notin \{A, B\}$. The modified test says yes: $\det(\{A, B\}) = \{A, B, V\}$, both parents of V being evidence, and deleting the edges out of A, B, V isolates C and D, so $\mathrm{dsep}_G(C, \{A, B, V\}, D)$ holds by Theorem 4.1. The independence is real: a, b pins $V = a \oplus b$, and C and D are independent given V, while marginally V is an unobserved common cause of both.

Problem 4.21. Let Pr be a probability distribution over variables $\mathbf{X}$ and let $\mathbf{B}$ be a Markov blanket for variable X. Show the correctness of the following procedure for finding a Markov boundary for X.

- Let $\mathbf{R}$ be $\mathbf{X} \setminus (\{X\} \cup \mathbf{B})$.
- Repeat until every variable in $\mathbf{B}$ has been examined or $\mathbf{B}$ is empty:
  1. Pick a variable Y in $\mathbf{B}$.
  2. Test whether $I_{\Pr}(X, \mathbf{B} \setminus \{Y\}, \mathbf{R} \cup \{Y\})$.
  3. If the test succeeds, remove Y from $\mathbf{B}$, add it to $\mathbf{R}$, and go to Step 1.
- Declare $\mathbf{B}$ a Markov boundary for X and exit.

Hint: Appeal to the weak union axiom.

Solution. Everything follows from one lemma: if $\mathbf{S}$ is a Markov blanket for X and $\mathbf{S} \subseteq \mathbf{C}$ with $X \notin \mathbf{C}$, then $\mathbf{C}$ is a Markov blanket too. For splitting the right-hand argument of $I_{\Pr}(X, \mathbf{S}, \mathbf{X} \setminus \mathbf{S} \setminus \{X\})$ disjointly as $(\mathbf{C} \setminus \mathbf{S}) \cup (\mathbf{X} \setminus \mathbf{C} \setminus \{X\})$ and applying weak union (4.7) yields

$$I_{\Pr}\big(X,\ \mathbf{C},\ \mathbf{X} \setminus \mathbf{C} \setminus \{X\}\big),$$

which with $X \notin \mathbf{C}$ is Definition 4.3 for $\mathbf{C}$.

Blanket. Write $\mathbf{B}_t, \mathbf{R}_t$ for the two sets at the start of iteration t, and take as invariant that $\mathbf{B}_t \cap \mathbf{R}_t = \emptyset$, $\mathbf{B}_t \cup \mathbf{R}_t \cup \{X\} = \mathbf{X}$, $X \notin \mathbf{B}_t$ and $\mathbf{B}_t$ is a Markov blanket for X. It holds initially, by the choice of $\mathbf{R}$ and the hypothesis on $\mathbf{B}$. If the Step 2 test succeeds for the chosen Y, then since $\mathbf{R}_t \cup \{Y\} = \mathbf{X} \setminus (\mathbf{B}_t \setminus \{Y\}) \setminus \{X\}$ by the invariant, the test is exactly Definition 4.3 for $\mathbf{B}_{t+1} = \mathbf{B}_t \setminus \{Y\}$, $\mathbf{R}_{t+1} = \mathbf{R}_t \cup \{Y\}$, and the set equations survive; if it fails nothing changes. So the set $\mathbf{B}^*$ returned on exit is a Markov blanket. (If $\mathbf{B}$ empties, $\emptyset$ is returned and is trivially a boundary.)

Minimality. Suppose some Markov blanket $\mathbf{S} \subsetneq \mathbf{B}^*$ and pick $Y \in \mathbf{B}^* \setminus \mathbf{S}$. Since the loop runs until every variable of $\mathbf{B}$ has been examined and Y survives, some iteration t picked Y and failed the test. There $\mathbf{S} \subseteq \mathbf{B}^* \setminus \{Y\} \subseteq \mathbf{B}_t \setminus \{Y\}$, the second inclusion because $\mathbf{B}$ only shrinks, and $X \notin \mathbf{B}_t \setminus \{Y\}$; so the lemma with $\mathbf{C} = \mathbf{B}_t \setminus \{Y\}$ gives

$$I_{\Pr}\big(X,\ \mathbf{B}_t \setminus \{Y\},\ \mathbf{X} \setminus (\mathbf{B}_t \setminus \{Y\}) \setminus \{X\}\big) = I_{\Pr}(X,\ \mathbf{B}_t \setminus \{Y\},\ \mathbf{R}_t \cup \{Y\}),$$

the identity again by the invariant. But that is the Step 2 test at iteration t, which failed. Hence no proper subset of $\mathbf{B}^*$ is a blanket, and $\mathbf{B}^*$ is a Markov boundary for X. ■

3.4 Exercises 4.22–4.25

Problem 4.22. Show that every probability distribution Pr over variables $X_1, \ldots, X_n$ can be induced by some Bayesian network (G, Θ) over variables $X_1, \ldots, X_n$. In particular, show how (G, Θ) can be constructed from Pr.

Solution. Let G be the complete DAG on the ordering $X_1, \ldots, X_n$, with $X_j \to X_i$ for every $j < i$, so that $\mathbf{U}_i = \{X_1, \ldots, X_{i-1}\}$ and $\mathbf{U}_1 = \emptyset$, and take

$$\theta_{x_i|\mathbf{u}_i} \;=\; \begin{cases} \Pr(x_i \mid \mathbf{u}_i), & \text{if } \Pr(\mathbf{u}_i) > 0, \\ 1/|X_i|, & \text{if } \Pr(\mathbf{u}_i) = 0. \end{cases}$$

Every edge increases the index of its head, so G is acyclic, and the parametrization is legal in the sense of Definition 4.1, each row summing to 1: a conditional distribution over X_i in the first case, uniform in the second (rows over a zero-probability parent instantiation are unconstrained by Pr). The parameters compatible with an instantiation $x_1, \ldots, x_n$ are $\theta_{x_i|x_1,\ldots,x_{i-1}}$, one per family, so the chain rule for Bayesian networks (4.2) gives

$$\Pr'(x_1, \ldots, x_n) \;=\; \prod_{i=1}^{n} \theta_{x_i|x_1,\ldots,x_{i-1}},$$

with $\Pr(x_1, \ldots, x_0) = 1$ for the empty prefix.
(i) $\Pr(x_1, \ldots, x_n) > 0$. The prefix events are nested, so every $\Pr(x_1, \ldots, x_{i-1}) \geq \Pr(x_1, \ldots, x_n) > 0$ and the first branch applies throughout:

$$\begin{aligned}\prod_{i=1}^{n} \theta_{x_i|x_1,\ldots,x_{i-1}} &= \prod_{i=1}^{n} \Pr(x_i \mid x_1, \ldots, x_{i-1}) \\ &= \prod_{i=1}^{n} \frac{\Pr(x_1, \ldots, x_i)}{\Pr(x_1, \ldots, x_{i-1})} = \Pr(x_1, \ldots, x_n),\end{aligned}$$

the product telescoping; this is the chain rule (3.16).
(ii) $\Pr(x_1, \ldots, x_n) = 0$. Let k be least with $\Pr(x_1, \ldots, x_k) = 0$. Then $\Pr(x_1, \ldots, x_{k-1}) > 0$, so the first branch applies to the k-th factor, $\theta_{x_k|x_1,\ldots,x_{k-1}} = \Pr(x_1, \ldots, x_k)/\Pr(x_1, \ldots, x_{k-1}) = 0$, and the whole product vanishes.
Hence $\Pr' = \Pr$. ■

Problem 4.23. Let G be a DAG and let G' be an undirected graph generated from G as follows:
1. For every node in G, every pair of its parents are connected by an undirected edge.
2. Every directed edge in G is converted into an undirected edge.

For every variable X, let B_X be its neighbors in G' and Z_X be all variables excluding X and B_X. Show that X and Z_X are d-separated by B_X in DAG G.

Solution. G' is the moral graph of G, so B_X is the set of parents, children and spouses of X (a spouse being a node sharing a child with X):

$$B_X \;=\; \Big(\mathrm{pa}(X) \cup \mathrm{ch}(X) \cup \bigcup_{C \in \mathrm{ch}(X)} \mathrm{pa}(C)\Big) \setminus \{X\},$$

an edge of G' at X arising either from step 2, as an edge of G into or out of X, or from step 1, with X and the other endpoint distinct parents of a common node. Acyclicity gives $X \notin B_X$, so $\{X\}$, B_X, Z_X are pairwise disjoint and $\mathrm{dsep}_G(\{X\}, B_X, Z_X)$ is well formed.
Suppose some path $\alpha : X = W_0 - W_1 - \cdots - W_m = Y$ with $Y \in Z_X$ is unblocked by B_X, so that every valve on it is open (Definition 4.2).
1. W_1 is adjacent to X, hence a parent or a child of X, hence in B_X; as $Y \notin B_X$ this gives $W_1 \neq Y$ and $m \geq 2$, so W_1 is internal and carries a valve. Being open at a node of the conditioning set, that valve is not sequential or divergent, so it is convergent: $X \to W_1 \leftarrow W_2$.
2. Path nodes are distinct, so $W_2 \neq X$ and X, W_2 are distinct parents of W_1; step 1 of the construction therefore puts the edge $X - W_2$ into G', giving $W_2 \in B_X$, hence $W_2 \neq Y$ and W_2 internal. Its edge to W_1 is $W_2 \to W_1$, outgoing, so its valve is sequential or divergent, and $W_2 \in B_X$ closes it.

That contradicts α being unblocked, so every path from X to a node of Z_X is blocked by B_X and $\mathrm{dsep}_G(\{X\}, B_X, Z_X)$.

Problem 4.24. Let G be a DAG and let $\mathbf{X}$, $\mathbf{Y}$, and $\mathbf{Z}$ be three disjoint sets of nodes in G. Let G' be an undirected graph constructed from G according to the following steps:
1. Every node is removed from G unless it is in $\mathbf{X} \cup \mathbf{Y} \cup \mathbf{Z}$ or one of its descendants is in $\mathbf{X} \cup \mathbf{Y} \cup \mathbf{Z}$.
2. For every node in G, every pair of its parents are connected by an undirected edge.
3. Every directed edge in G is converted into an undirected edge.

Show that $\mathrm{dsep}_G(\mathbf{X}, \mathbf{Z}, \mathbf{Y})$ if and only if $\mathbf{X}$ and $\mathbf{Y}$ are separated by $\mathbf{Z}$ in G' (i.e., every path between $\mathbf{X}$ and $\mathbf{Y}$ in G' must pass through $\mathbf{Z}$).

Solution. We prove both directions in contrapositive form. Write $\mathrm{An}(\mathbf{S})$ for $\mathbf{S}$ together with all ancestors of its members; step 1 retains exactly $\mathbf{A} = \mathrm{An}(\mathbf{X} \cup \mathbf{Y} \cup \mathbf{Z})$, which is ancestrally closed, so a retained

node keeps all its parents, step 2 marries exactly the pairs of distinct parents in G of retained nodes, and step 3 keeps every G-edge with both endpoints retained. By Definition 4.2 a path is **active** given $\mathbf{Z}$ when all its valves are open: a sequential or divergent valve at W iff $W \notin \mathbf{Z}$, a convergent one iff $W \in \mathrm{An}(\mathbf{Z})$. So $\mathrm{dsep}_G(\mathbf{X}, \mathbf{Z}, \mathbf{Y})$ fails exactly when some $\mathbf{X}$-$\mathbf{Y}$ path is active.

Lemma 1. Every node of an active $\mathbf{X}$-$\mathbf{Y}$ path α lies in $\mathbf{A}$, hence survives step 1. The endpoints do. For internal W: if its valve is convergent it is open, so $W \in \mathrm{An}(\mathbf{Z})$. Otherwise W has an edge on α outgoing from it; walk that way. Each node reached is entered along an edge directed into it, so its valve is sequential (and we leave along its outgoing edge) or convergent. Either we reach an endpoint by a directed path from W, giving $W \in \mathrm{An}(\mathbf{X} \cup \mathbf{Y})$, or we reach by a directed path the first node K with a convergent valve, which is open, giving $K \in \mathrm{An}(\mathbf{Z})$ and so $W \in \mathrm{An}(\mathbf{Z})$. (This is the argument in the book's proof of Theorem 4.1.) □

Lemma 2. Adjacent internal nodes of a path cannot both carry convergent valves, the edge between them having to be directed into each. □

Direction 1: d-separation failing in G means separation failing in G'. Let $\alpha : X = W_0, \ldots, W_m = Y$ be active with $X \in \mathbf{X}$, $Y \in \mathbf{Y}$; by Lemma 1 all its nodes are retained. Delete from α every internal node with a convergent valve, obtaining σ. By Lemma 2 the deleted nodes are pairwise non-adjacent on α, so consecutive nodes U, V of σ are either adjacent on α, whence $U - V$ is an edge of G' by step 3, or separated by one deleted W with $U \to W \leftarrow V$, whence $U - V$ is an edge of G' by step 2, U and V being distinct parents of the retained W. Its nodes being a sublist of those of α, σ is a path of G' from X to Y, and it avoids $\mathbf{Z}$: the endpoints lie in $\mathbf{X} \cup \mathbf{Y}$ and each surviving internal node carries an open sequential or divergent valve.

Direction 2: separation failing in G' means d-separation failing in G. Work with walks (consecutive nodes adjacent, repetitions allowed), valves being defined at internal positions as for paths. Call a walk $\pi = W_0, \ldots, W_m$ **admissible** when (i) one endpoint lies in $\mathbf{X}$ and the other in $\mathbf{Y}$; (ii) every node lies in $\mathbf{A}$; (iii) every sequential or divergent valve on π is open. By (iii) a closed valve on an admissible walk is convergent at some $U \notin \mathrm{An}(\mathbf{Z})$; call it **bad**.

Step A: from a G'-path to an admissible walk. Let $\sigma : V_0, \ldots, V_k$ be a G'-path with $V_0 \in \mathbf{X}$, $V_k \in \mathbf{Y}$, no $V_i \in \mathbf{Z}$. Each edge of σ came from step 3, so is an edge of G, or from step 2, so joins distinct parents of some retained C; replace each edge of the second kind by $V_i \to C \leftarrow V_{i+1}$. The resulting walk π satisfies (i); its nodes are the V_i, which lie in $\mathbf{A}$ as nodes of G', and the inserted retained C, giving (ii); and each inserted C carries a convergent valve, so every sequential or divergent valve sits at some $V_i \notin \mathbf{Z}$, giving (iii).

Step B: an admissible walk with no bad valve exists. Induct on the number b of bad valves; for $b > 0$ let $U = W_i$ carry the leftmost, so $W_{i-1} \to W_i \leftarrow W_{i+1}$ and the valves at $1, \ldots, i-1$ are open. Since $U \notin \mathrm{An}(\mathbf{Z})$ while $U \in \mathbf{A}$, take a shortest directed path

$$U \to D_1 \to \cdots \to D_t = D, \qquad t \geq 0,$$

to a node $D \in \mathbf{X} \cup \mathbf{Y} \cup \mathbf{Z}$ ($t = 0$ meaning $D = U$). As $\mathbf{Z} \subseteq \mathrm{An}(\mathbf{Z})$ we get $D \in \mathbf{X} \cup \mathbf{Y}$, so $U, D \notin \mathbf{Z}$ by disjointness, minimality puts $D_1, \ldots, D_{t-1}$ outside $\mathbf{X} \cup \mathbf{Y} \cup \mathbf{Z}$, and every D_l, an ancestor of D, lies in $\mathbf{A}$. Consider

$$\pi_1 = W_0, \ldots, W_i, D_1, \ldots, D_t,$$
$$\pi_2 = D_t, \ldots, D_1, W_i, W_{i+1}, \ldots, W_m.$$

In π_1 the valves at $W_1, \ldots, W_{i-1}$ are those of π, hence open; the valve at W_i is $W_{i-1} \to W_i \to D_1$, sequential with $W_i \notin \mathbf{Z}$; and those at $D_1, \ldots, D_{t-1}$ are sequential outside $\mathbf{Z}$. So π_1 satisfies (ii), (iii) and has no bad valve. In π_2 the valves at $D_{t-1}, \ldots, D_1$ are sequential outside $\mathbf{Z}$, that at W_i is $W_{i+1} \to W_i \to D_1$, sequential and open, and those at $W_{i+1}, \ldots, W_{m-1}$ are the valves of π there; so π_2 satisfies (ii), (iii) with exactly $b - 1$ bad valves. (For $t = 0$, W_i becomes an endpoint carrying no valve and the checks at W_i and the D_l are simply absent.) If D lies in whichever of $\mathbf{X}, \mathbf{Y}$ does not contain W_0, then π_1 satisfies (i) and we are done; otherwise D lies in the set not containing W_m, so π_2 satisfies (i) and the induction applies. Call an admissible walk with no bad valve **active**: all its valves are open.

Step C: an active walk yields an active path. Pick an active walk $\pi = W_0, \ldots, W_m$ of minimum length

with one endpoint in $\mathbf{X}$ and the other in $\mathbf{Y}$, and suppose $W_i = W_j$ with $i < j$. The shorter walk

$$\pi^* = W_0, \ldots, W_i, W_{j+1}, \ldots, W_m$$

has the same endpoints, and every valve of it other than the one at position i is a valve of π, hence open. If $i = 0$ or $j = m$, position i is an endpoint and no new valve arises. Otherwise let e join W_{i-1} to W_i and f join W_j to W_{j+1}.

- (i) e outgoing from W_i (or, symmetrically, f outgoing from $W_j = W_i$): then the valve of π at i (at j) shares that outgoing edge, so it is sequential or divergent and, being open, gives $W_i \notin \mathbf{Z}$; the new valve also has an outgoing edge, so it too is sequential or divergent, hence open.
- (ii) e and f both into W_i: the new valve is convergent, open iff $W_i \in \mathrm{An}(\mathbf{Z})$. If the valve of π at i or at j is convergent, it is open and gives exactly that. Otherwise both are sequential, $W_{i-1} \to W_i \to W_{i+1}$ and $W_{j+1} \to W_j \to W_{j-1}$. Traverse the closed walk $W_i, W_{i+1}, \ldots, W_j = W_i$: its first edge runs along the traversal and its last against it, so there is a least r with $i < r < j$ and $W_{r+1} \to W_r$. By minimality $W_i \to W_{i+1} \to \cdots \to W_r$ is directed and the valve of π at W_r is convergent, hence open, so $W_r \in \mathrm{An}(\mathbf{Z})$ and W_i, its ancestor, lies in $\mathrm{An}(\mathbf{Z})$ too.

Either way π^* is a shorter active walk with the same endpoints, contradicting minimality. So the minimal active walk is a path: an unblocked $\mathbf{X}$-$\mathbf{Y}$ path, and $\mathrm{dsep}_G(\mathbf{X}, \mathbf{Z}, \mathbf{Y})$ fails. Chaining Steps A, B, C gives the contrapositive, and with Direction 1,

$$\mathrm{dsep}_G(\mathbf{X}, \mathbf{Z}, \mathbf{Y}) \iff \mathbf{X} \text{ and } \mathbf{Y} \text{ are separated by } \mathbf{Z} \text{ in } G'.$$

Problem 4.25. Let X and Y be two nodes in a DAG G that are not connected by an edge. Let $\mathbf{Z}$ be a set of nodes defined as follows: $Z \in \mathbf{Z}$ if and only if $Z \notin \{X, Y\}$ and Z is an ancestor of X or an ancestor of Y. Show that $\mathrm{dsep}_G(X, \mathbf{Z}, Y)$.

Solution. By the definition of $\mathbf{Z}$,

$$\{X, Y\} \cup \mathbf{Z} \;=\; \mathrm{An}(\{X, Y\}), \qquad \mathbf{Z} \;=\; \mathrm{An}(\{X, Y\}) \setminus \{X, Y\},$$

writing $\mathrm{An}(\mathbf{S})$ for $\mathbf{S}$ with all ancestors of its members; the three sets $\{X\}$, $\{Y\}$, $\mathbf{Z}$ are pairwise disjoint, so the statement is well formed, and $\mathrm{An}(\{X, Y\})$ being ancestrally closed gives $\mathrm{An}(\mathbf{Z}) \subseteq \{X, Y\} \cup \mathbf{Z}$. Suppose some path $\alpha : X = W_0 - W_1 - \cdots - W_m = Y$ is unblocked by $\mathbf{Z}$. Lemma 1 of Exercise 4.24 places every node of α in $\mathrm{An}(\{X, Y\} \cup \mathbf{Z}) = \mathrm{An}(\{X, Y\})$, and path nodes are distinct, so every internal node of α lies in $\mathrm{An}(\{X, Y\}) \setminus \{X, Y\} = \mathbf{Z}$ and hence carries a convergent valve, a sequential or divergent valve at a node of $\mathbf{Z}$ being closed. By Lemma 2 of Exercise 4.24 adjacent internal nodes cannot both be convergent, so α has at most one internal node, and it has at least one since X and Y are not joined by an edge. Hence $m = 2$ and α is

$$X \to W_1 \leftarrow Y,$$

with $W_1 \in \mathbf{Z}$ an ancestor of X or of Y; the directed path from W_1 to that node closes a cycle with the edge $X \to W_1$ or $Y \to W_1$, against the acyclicity of G. So no path is unblocked and $\mathrm{dsep}_G(X, \mathbf{Z}, Y)$. Method (2): in Exercise 4.24 the retained set is $\mathbf{A} = \{X, Y\} \cup \mathbf{Z}$, so a G'-path from X to Y avoiding $\mathbf{Z}$ has no internal node, i.e. X and Y are adjacent in G'. That needs X and Y adjacent in G, excluded, or both parents of some retained $C \in \mathbf{Z}$, which is an ancestor of X or Y and again closes a directed cycle. So $\mathbf{Z}$ separates X from Y in G', and Exercise 4.24 applies.

4 Building Bayesian Networks

Contents

4.1 Exercises 5.1–5.7 49
4.2 Exercises 5.8–5.14 59
4.3 Exercises 5.15–5.16 66

4.1 Exercises 5.1–5.7

Problem 5.1. Joe's x-ray test comes back positive for lung cancer. The test's false negative rate is $f_n = .40$ and its false positive rate is $f_p = .02$. We also know that the prior probability of having lung cancer is $c = .001$. Describe a Bayesian network and a corresponding query for computing the probability that Joe has lung cancer given his positive x-ray. What is the value of this probability? Use sensitivity analysis to identify necessary and sufficient conditions on each of f_n, f_p, and c that guarantee the probability of cancer to be no less than 10% given a positive x-ray test.

Solution. Network. Two binary variables: the query variable C with values yes and no, and the evidence variable T with values $+$ and $-$. Cancer is the only direct cause named, so the structure is the single edge

$$C \longrightarrow T.$$

The CPTs are read off the statement, the prior for C being the prevalence c:

C	θ_C
yes	$c = .001$
no	.999

The CPT for T is the test specification, a false negative being a $-$ on a cancer patient and a false positive a $+$ on a healthy one:

C	T	$\theta_{T\mid C}$
yes	$+$	$1 - f_n = .60$
yes	$-$	$f_n = .40$
no	$+$	$f_p = .02$
no	$-$	$1 - f_p = .98$

Query. The posterior marginal $\Pr(C = \text{yes} \mid T = +)$, with evidence $T = +$.
Value. By the chain rule for Bayesian networks $\Pr(C, T) = \theta_C\, \theta_{T\mid C}$, so

$$\begin{aligned}\Pr(C = \text{yes} \mid T = +) &= \frac{c\,(1 - f_n)}{c\,(1 - f_n) + (1 - c)\,f_p} \\ &= \frac{(.001)(.60)}{(.001)(.60) + (.999)(.02)} \\ &= \frac{.0006}{.0006 + .01998} = \frac{.0006}{.02058} = \frac{10}{343}.\end{aligned}$$

that is, $\Pr(C = \text{yes} \mid T = +) \approx 2.92\%$.
Sensitivity analysis. Clearing the denominator, which is positive whenever the evidence has nonzero probability, gives the necessary and sufficient condition

$$\frac{c(1 - f_n)}{c(1 - f_n) + (1 - c)f_p} \;\geq\; \frac{1}{10} \iff 9\,c\,(1 - f_n) \;\geq\; (1 - c)\,f_p,$$

and solving it for one parameter with the other two at their stated values:

- f_p: $.0054 \geq .999\, f_p$, so $f_p \leq 1/185 \approx .0054$, a fourfold improvement.
- c: $5.42\, c \geq .02$, so $c \geq 1/271 \approx .00369$, nearly four times the prevalence.
- f_n: $.009\,(1 - f_n) \geq .01998$, so $1 - f_n \geq 111/50 = 2.22$, impossible. No value of f_n works; even $f_n = 0$ gives only $.001/(.001 + .01998) \approx 4.77\%$.

Problem 5.2. We have three identical and independent temperature sensors that will trigger in:

- 90% of the cases where the temperature is high
- 5% of the cases where the temperature is nominal
- 1% of the cases where the temperature is low.

The probability of high temperature is 20%, nominal temperature is 70%, and low temperature is

10%. Describe a Bayesian network and corresponding queries for computing the following:
- (a) Probability that the first sensor will trigger given that the other two sensors have also triggered
- (b) Probability that the temperature is high given that all three sensors have triggered
- (c) Probability that the temperature is high given that at least one sensor has triggered

Solution. Network. Let T have values high, nominal, low and let S_1, S_2, S_3 be binary with values on and off. The temperature is the only direct cause of each reading and the sensors are independent given it, so the structure is the naive Bayes star

$$S_1 \longleftarrow T \longrightarrow S_2, \qquad T \longrightarrow S_3,$$

that is, T is a root and each S_i has T as its only parent. The CPTs are

T	θ_T
high	.20
nominal	.70
low	.10

and, identically for $i = 1, 2, 3$,

T	S_i	$\theta_{S_i \mid T}$
high	on	.90
nominal	on	.05
low	on	.01

(the rows for $S_i =$ off are the complements and are omitted as redundant).
For (c) the event "at least one sensor triggered" is a disjunction, the value of no variable of the network, so by the technique of Section 5.3.1 add a binary auxiliary variable O with parents S_1, S_2, S_3 and the deterministic or-gate CPT

$$\Pr(O = \text{true} \mid s_1, s_2, s_3) = \begin{cases} 1, & \text{if some } s_i = \text{on}, \\ 0, & \text{otherwise}, \end{cases}$$

turning the disjunctive evidence into the ordinary evidence $O =$ true.
Write $\alpha_t = \theta_t$ and $\sigma_t = \Pr(S_i = \text{on} \mid t)$; the S_i being independent given T,

$$\Pr(s_1, s_2, s_3) = \sum_t \alpha_t \prod_i \Pr(s_i \mid t).$$

(a) The query is $\Pr(S_1 = \text{on} \mid S_2 = \text{on}, S_3 = \text{on})$, and S_1 is not independent of S_2, S_3, the unobserved common parent T leaving $S_1 \leftarrow T \rightarrow S_2$ unblocked (Definition 4.2). Two sums are needed:

$$\begin{aligned} \Pr(S_2, S_3 \text{ on}) &= \sum_t \alpha_t \sigma_t^2 \\ &= (.2)(.81) + (.7)(.0025) + (.1)(.0001) \\ &= .162 + .00175 + .00001 \;=\; .16376, \end{aligned}$$

$$\begin{aligned} \Pr(S_1, S_2, S_3 \text{ on}) &= \sum_t \alpha_t \sigma_t^3 \\ &= (.2)(.729) + (.7)(.000125) + (.1)(.000001) \\ &= .1458 + .0000875 + .0000001 \;=\; .1458876. \end{aligned}$$

Hence

$$\Pr(S_1 = \text{on} \mid S_2, S_3 \text{ on}) = \frac{.1458876}{.16376} = \frac{364719}{409400} \approx .8909.$$

Two triggers raise the chance that the third fires from its prior $\sum_t \alpha_t \sigma_t = .216$ to about 89.1%.
(b) The query is $\Pr(T = \text{high} \mid S_1, S_2, S_3 \text{ on})$, and the numerator terms are already computed:

$$\Pr(T = \text{high} \mid S_1, S_2, S_3 \text{ on}) = \frac{.1458}{.1458876} = \frac{121500}{121573} \approx .99940.$$

(c) The query is $\Pr(T = \text{high} \mid O = \text{true})$, and conditional independence of the sensors gives $\Pr(O = \text{true} \mid t) = 1 - (1 - \sigma_t)^3$:

T	α_t	$1 - (1 - \sigma_t)^3$	$\alpha_t\,[1 - (1 - \sigma_t)^3]$
high	.20	.999	.1998
nominal	.70	.142625	.0998375
low	.10	.029701	.0029701

Summing the last column, $\Pr(O = \text{true}) = .3026076$, so

$$\Pr(T = \text{high} \mid O = \text{true}) = \frac{.1998}{.3026076} = \frac{166500}{252173} \approx .6603.$$

Problem 5.3. Suppose that we apply three test vectors to the circuit in Figure 5.14, where each gate is initially ok with probability .99. As we change the test vector, a gate that is ok may become faulty with probability .01, and a gate that is faulty may become ok with probability .001.
The circuit of Figure 5.14(a) has primary inputs A and B and primary output E. It contains three gates: an inverter X whose input is A and whose output is the internal wire C; an and-gate Y whose inputs are A and B and whose output is the internal wire D; and an or-gate Z whose inputs are C and D and whose output is E. The Bayesian network of Figure 5.14(b) accordingly has roots A, B, X, Y, Z and edges

$$A \to C, \quad X \to C, \quad A \to D, \quad B \to D, \quad Y \to D,$$
$$C \to E, \quad D \to E, \quad Z \to E.$$

Wires take values low / high and health variables take values ok / faulty. A gate that is ok computes its logical function; a gate that is faulty outputs high with probability .5 regardless of its inputs. The primary inputs are uniform.

- (a) What are the posterior marginals for the health variables given the following test vectors?
  - $A = \text{high}$, $B = \text{high}$, $E = \text{low}$
  - $A = \text{low}$, $B = \text{low}$, $E = \text{high}$
  - $A = \text{low}$, $B = \text{high}$, $E = \text{high}$
- (b) What about the following test vectors?
  - $A = \text{high}$, $B = \text{high}$, $E = \text{low}$
  - $A = \text{low}$, $B = \text{low}$, $E = \text{high}$
  - $A = \text{low}$, $B = \text{high}$, $E = \text{low}$
- (c) What are the MPE and MAP (over health variables) for each of the cases in (a) and (b)?

Assume that the test vectors are applied in the order given.

Solution. Network. Faults being intermittent, use the dynamic Bayesian network of Figure 5.15(a) over three time slices; slice $t \in \{1, 2, 3\}$ holds a full copy of the circuit network,

$$A_t,\ B_t,\ X_t,\ Y_t,\ Z_t,\ C_t,\ D_t,\ E_t,$$

with the intra-slice edges listed in the statement, and consecutive slices are linked by the persistence edges $X_t \to X_{t+1}$, $Y_t \to Y_{t+1}$, $Z_t \to Z_{t+1}$. The health CPTs are the prior in slice 1 and the persistence model in slices 2 and 3 (identically for Y and Z):

X_1	θ
ok	.99
faulty	.01

X_t	X_{t+1}	θ
ok	ok	.99
ok	faulty	.01
faulty	ok	.001
faulty	faulty	.999

The gate CPTs are the two-valued ones of Section 5.3.5, for example

A	X	C	$\theta_{c\mid a,x}$
high	ok	high	0
low	ok	high	1
high	faulty	high	.5
low	faulty	high	.5

and analogously $\Pr(D = \text{high} \mid a, b, Y = \text{ok}) = [\![a \wedge b]\!]$, $\Pr(D = \text{high} \mid a, b, Y = \text{faulty}) = .5$, and $\Pr(E = \text{high} \mid c, d, Z = \text{ok}) = [\![c \vee d]\!]$, $\Pr(E = \text{high} \mid c, d, Z = \text{faulty}) = .5$.

Reduction. Write $h_t = (x_t, y_t, z_t)$ for the health state in slice t and $e_t = (a_t, b_t, \varepsilon_t)$ for the t-th test vector. The inputs A_t, B_t are roots d-separated from the health variables, so as in Section 5.3.5 their CPTs cancel and

$$\Pr(h_1, h_2, h_3 \mid e_1, e_2, e_3) \;\propto\; \Pr(h_1)\Pr(h_2 \mid h_1)\Pr(h_3 \mid h_2)\prod_{t=1}^{3} L_t(h_t),$$

where $L_t(h) = \Pr(\varepsilon_t \mid a_t, b_t, h)$ comes from eliminating C_t and D_t. Rows below are (X, Y, Z) with o = ok, f = faulty; columns are the test vectors, *hhl* abbreviating $A = \text{high}, B = \text{high}, E = \text{low}$:

(X, Y, Z)	*hhl*	*llh*	*lhh*	*lhl*
o,o,o	0	1	1	0
o,o,f	1/2	1/2	1/2	1/2
o,f,o	1/2	1	1	0
o,f,f	1/2	1/2	1/2	1/2
f,o,o	0	1/2	1/2	1/2
f,o,f	1/2	1/2	1/2	1/2
f,f,o	1/4	3/4	3/4	1/4
f,f,f	1/2	1/2	1/2	1/2

The zero entries check out: under *hhl*, Y ok forces $D = \text{high}$ and then Z ok forces $E = \text{high}$ whatever C is, killing rows (o,o,o) and (f,o,o); under *lhl*, X ok gives $C = \text{high}$ and Z ok forces $E = \text{high}$, killing every row with X and Z both ok. Summing the $8^3 = 512$ terms gives normalizing constants $\Pr(\varepsilon_1, \varepsilon_2, \varepsilon_3 \mid a, b) \approx 6.0751 \times 10^{-3}$ for (a) and $\approx 1.3097 \times 10^{-3}$ for (b).

(a) Posterior marginals.

variable	$\Pr(\cdot = \text{faulty} \mid e)$
X_1	.00433
Y_1	.79551
Z_1	.20655
X_2	.01086
Y_2	.79677
Z_2	.20778
X_3	.01879
Y_3	.79801
Z_3	.21137

The inverter X is cleared and the blame falls on Y ($\approx 80\%$) and Z ($\approx 21\%$): only the first vector is abnormal, and the Y explanation costs nothing on the other two ($L = 1$ at (o,f,o)) while the Z explanation pays $1/2$ on each, the $4:1$ odds ratio the marginals reflect.

(b) Posterior marginals.

variable	$\Pr(\cdot = \text{faulty} \mid e)$
X_1	.01313
Y_1	.05691
Z_1	.95262
X_2	.02968
Y_2	.06627
Z_2	.96195
X_3	.04840
Y_3	.07552
Z_3	.98049

The third vector cannot be explained by a faulty Y at all, X ok making C high and an ok Z then forcing E high, so a Y-based story needs a second fault, whereas one persistent fault in Z covers both abnormal vectors.

(c) MPE and MAP. MAP is taken over the nine health variables $X_1, Y_1, Z_1, \ldots, X_3, Y_3, Z_3$; MPE is taken over all non-evidence variables, i.e. the health variables together with the internal wires C_t, D_t.

Case (a). The MAP answer is

slice	X	Y	Z
1	ok	faulty	ok
2	ok	faulty	ok
3	ok	faulty	ok

with $\Pr(\text{MAP} \mid e) \approx 77.33\%$, a persistent and-gate fault; the runner-up, X, Y ok and Z faulty throughout, sits at $\approx 19.33\%$, a factor 4 behind through the $(1/2)^2$ penalty the Z story pays on the two normal vectors.

The MPE is not a completion of the MAP answer and is not unique: its probability is $\approx 19.33\%$, with five tied instantiations. One extends the Z-fault hypothesis,

$$X_t = Y_t = \text{ok},\ Z_t = \text{faulty}\ (t = 1, 2, 3),$$
$$(C_1, D_1) = (\text{low}, \text{high}), \quad (C_2, D_2) = (C_3, D_3) = (\text{high}, \text{low}),$$

and the other four extend the Y-fault hypothesis,

$$X_t = Z_t = \text{ok},\ Y_t = \text{faulty}\ (t = 1, 2, 3),$$
$$(C_1, D_1) = (\text{low}, \text{low}), \quad C_2 = C_3 = \text{high},$$

with D_2 and D_3 free, four equally likely combinations, a faulty Y making D a coin flip while $C = \text{high}$ already forces $E = \text{high}$. The Y hypothesis wins the MAP by spreading its mass over four wire completions, each only as likely as the single Z-fault completion, which is why the Z hypothesis's MAP and the MPE coincide at 19.33%.

Case (b). MAP and MPE agree. The MAP answer is

slice	X	Y	Z
1	ok	ok	faulty
2	ok	ok	faulty
3	ok	ok	faulty

with $\Pr(\text{MAP} \mid e) \approx 89.68\%$, and the MPE is its unique completion

$$(C_1, D_1) = (\text{low}, \text{high}), \qquad (C_2, D_2) = (C_3, D_3) = (\text{high}, \text{low}),$$

also at $\approx 89.68\%$ and with no ties. The runner-up is far behind: Y faulty throughout with Z also faulty in slice 3, at $\approx 1.81\%$, the double fault the third test vector forces on any Y-based explanation.

Problem 5.4. We have two sensors that are meant to detect extreme temperature, which occurs 20% of the time. The sensors have identical specifications with a false positive rate of 1% and a false negative rate of 3%. If the power is off (dead battery), the sensors will read negative regardless of the temperature. Suppose now that we have two sensor kits: Kit A where both sensors receive power from the same battery and Kit B where they receive power from independent batteries. Assuming that each battery has a .9 probability of power availability, what is the probability of extreme temperature given each of the following scenarios:

- (a) The two sensors read negative
- (b) The two sensors read positive
- (c) One sensor reads positive while the other reads negative.

Answer the previous questions with respect to each of the two kits.

Solution. Networks. Let T have values extreme and normal with $\Pr(T = \text{extreme}) = .2$ and let S_1, S_2 be the readings, values + and −. Each reading has two direct causes, the temperature and the power driving it, and the kits differ only in the number of battery variables.

Kit A (shared battery) has one root B with $\Pr(B = \text{on}) = .9$ and structure

$$S_1 \longleftarrow T \longrightarrow S_2,$$
$$S_1 \longleftarrow B \longrightarrow S_2.$$

Kit B (independent batteries) has two roots B_1, B_2, each with $\Pr(B_i = \text{on}) = .9$, and structure

$$S_1 \longleftarrow T \longrightarrow S_2, \qquad B_1 \to S_1, \qquad B_2 \to S_2.$$

The sensor CPT is the same in both kits, the specifications being identical: a false positive is a + on a normal temperature, a false negative a − on an extreme one, and a dead battery forces −:

T	battery	S_i	θ
extreme	on	+	.97
normal	on	+	.01
extreme	off	+	0
normal	off	+	0

The queries are the posterior marginals $\Pr(T = \text{extreme} \mid s_1, s_2)$; for (c) take $S_1 = +$, $S_2 = -$, the other ordering giving the same posterior by symmetry.

Kit A. Case on B, the only unobserved root besides T: with B on the readings are independent given T, with B off both are − with probability 1.

(a) Both negative. With the battery on a negative costs .03 under an extreme temperature and .99 under a normal one:

$$\Pr(-, -, \text{extreme}) = .2\,[\,.9(.03)^2 + .1\,] = .2(.10081) = .020162,$$
$$\Pr(-, -, \text{normal}) = .8\,[\,.9(.99)^2 + .1\,] = .8(.98209) = .785672,$$

so $\Pr(-, -) = .805834$ and

$$\Pr(T = \text{extreme} \mid -, -) = \frac{.020162}{.805834} = \frac{593}{23701} \approx .0250.$$

(b) Both positive. A + is impossible with a dead battery, so $B =$ off drops out:

$$\Pr(+, +, \text{extreme}) = .2(.9)(.97)^2 = .169362,$$
$$\Pr(+, +, \text{normal}) = .8(.9)(.01)^2 = .000072,$$

$$\Pr(T = \text{extreme} \mid +, +) = \frac{.169362}{.169434} = \frac{9409}{9413} \approx .99958.$$

(c) Mixed. The positive reading again forces $B =$ on:

$$\Pr(+, -, \text{extreme}) = .2(.9)(.97)(.03) = .005238,$$
$$\Pr(+, -, \text{normal}) = .8(.9)(.01)(.99) = .007128,$$

$$\Pr(T = \text{extreme} \mid +, -) = \frac{.005238}{.012366} = \frac{97}{229} \approx .4236.$$

Kit B. The roots B_1, B_2 being separate, fold each battery into its own sensor; given T the readings are independent, with effective per-sensor probabilities

$$\Pr(+ \mid \text{extreme}) = .9(.97) = .873, \quad \Pr(- \mid \text{extreme}) = .127,$$
$$\Pr(+ \mid \text{normal}) = .9(.01) = .009, \quad \Pr(- \mid \text{normal}) = .991.$$

(a) Both negative.

$$\Pr(-, -, \text{extreme}) = .2(.127)^2 = .0032258,$$
$$\Pr(-, -, \text{normal}) = .8(.991)^2 = .7856648,$$

$$\Pr(T = \text{extreme} \mid -, -) = \frac{.0032258}{.7888906} = \frac{16129}{3944453} \approx .00409.$$

(b) Both positive.

$$\Pr(+,+,\text{extreme}) = .2(.873)^2 = .1524258,$$
$$\Pr(+,+,\text{normal}) = .8(.009)^2 = .0000648,$$
$$\Pr(T = \text{extreme} \mid +,+) = \frac{.1524258}{.1524906} = \frac{9409}{9413} \approx .99958.$$

(c) Mixed.

$$\Pr(+,-,\text{extreme}) = .2(.873)(.127) = .0221742,$$
$$\Pr(+,-,\text{normal}) = .8(.009)(.991) = .0071352,$$
$$\Pr(T = \text{extreme} \mid +,-) = \frac{.0221742}{.0293094} = \frac{12319}{16283} \approx .7566.$$

Summary.

scenario	Kit A (shared battery)	Kit B (independent batteries)
(a) $--$	.0250	.00409
(b) $++$	.99958	.99958
(c) $+-$	.4236	.7566

Case (b) agrees across kits because a positive reading entails a live battery, so both networks reduce to the same naive Bayes model with sensor probabilities .97 and .01, giving $.2(.97)^2/[.2(.97)^2 + .8(.01)^2] = 9409/9413$. Under $-,-$ Kit A keeps about six times as much belief in an extreme temperature, the shared battery supplying one common-cause explanation of both negatives where Kit B would need two failures at $(.1)^2$; under $+,-$ the comparison reverses, the positive reading proving Kit A's shared battery live while saying nothing about Kit B's other battery.

Problem 5.5. Jack has three coins C_1, C_2, and C_3 with p_1, p_2, and p_3 as their corresponding probabilities of landing heads. Jack flips coin C_1 twice and then decides, based on the outcome, whether to flip coin C_2 or C_3 next. In particular, if the two C_1 flips come out the same, Jack flips coin C_2 three times next. However, if the C_1 flips come out different, he flips coin C_3 three times next. Given the outcome of Jack's last three flips, we want to know whether his first two flips came out the same. Describe a Bayesian network and a corresponding query that solves this problem. What is the solution to this problem assuming that $p_1 = .4$, $p_2 = .6$, and $p_3 = .1$ and the last three flips came out as follows:

- (a) tails, heads, tails
- (b) tails, tails, tails

Solution. Network. Take five flip variables $F_1, \ldots, F_5$ with values h and t, where F_1, F_2 are the C_1 flips and F_3, F_4, F_5 the last three, plus an auxiliary variable S with values same and different recording the comparison that governs Jack's decision (the device of Section 5.3.1, turning an event that is the value of no natural variable into one that can be queried). The causal story gives

$$F_1 \to S \leftarrow F_2, \qquad S \to F_3, \quad S \to F_4, \quad S \to F_5.$$

F_1 and F_2 are roots with $\Pr(F_i = h) = p_1 = .4$; S is a deterministic child of F_1, F_2 (an equivalence gate),

F_1	F_2	S	θ
h	h	same	1
h	t	same	0
t	h	same	0
t	t	same	1

and each of F_3, F_4, F_5 has S as its only parent, the value of S selecting which coin is being flipped:

S	F_j	θ
same	h	$p_2 = .6$
different	h	$p_3 = .1$

Query. The posterior marginal $\Pr(S = \text{same} \mid F_3, F_4, F_5)$, whose prior is

$$\Pr(S = \text{same}) = p_1^2 + (1 - p_1)^2 = .16 + .36 = .52.$$

The flips being independent given S, the likelihoods are products of coin parameters; writing L_2, L_3 for the likelihood of the observed triple under C_2 and under C_3,

$$\Pr(S = \text{same} \mid e) = \frac{.52\, L_2}{.52\, L_2 + .48\, L_3}.$$

(a) t, h, t.

$$L_2 = (1 - p_2)\, p_2\, (1 - p_2) = (.4)(.6)(.4) = .096,$$
$$L_3 = (1 - p_3)\, p_3\, (1 - p_3) = (.9)(.1)(.9) = .081.$$

Hence $\Pr(e, \text{same}) = (.52)(.096) = .04992$ and $\Pr(e, \text{different}) = (.48)(.081) = .03888$, so $\Pr(e) = .0888$ and

$$\Pr(S = \text{same} \mid e) = \frac{.04992}{.0888} = \frac{104}{185} \approx .5622.$$

The evidence is mild: $L_2/L_3 = 32/27$, so the posterior barely moves from the prior .52.
(b) t, t, t.

$$L_2 = (.4)^3 = .064, \qquad L_3 = (.9)^3 = .729.$$

Hence $\Pr(e, \text{same}) = (.52)(.064) = .03328$, $\Pr(e, \text{different}) = (.48)(.729) = .34992$, so $\Pr(e) = .3832$ and

$$\Pr(S = \text{same} \mid e) = \frac{.03328}{.3832} = \frac{208}{2395} \approx .0868.$$

Here $L_3/L_2 \approx 11.4$ favours the tails-biased C_3, flipped only when the first two flips differ, so the posterior on "same" drops from 52% to under 9%.

Problem 5.6. Lisa is given a fair coin C_1 and asked to flip it eight times in a row. Lisa also has a biased coin C_2 with a probability .8 of landing heads. All we know is that Lisa flipped the fair coin initially but we believe that she intends to switch to the biased coin and that she tends to be 10% successful in performing the switch. Suppose that we observe the outcome of the eight coin flips and want to find out whether Lisa managed to perform a coin switch and when. Describe a Bayesian network and a corresponding query that solves this problem. What is the solution to this problem assuming that the flips came out as follows:

- (a) tails, tails, tails, heads, heads, heads, heads, heads
- (b) tails, tails, heads, heads, heads, heads, heads, heads

Solution. Network. A dynamic Bayesian network with eight slices: S_i is the coin in Lisa's hand at flip i, with values fair and biased, and F_i the outcome, values h and t. The structure is the chain

$$S_1 \to S_2 \to \cdots \to S_8, \qquad S_i \to F_i \;\; (i = 1, \ldots, 8),$$

a hidden Markov model. The CPTs encode the three stated facts. Lisa flipped the fair coin initially:

S_1	θ
fair	1
biased	0

Between consecutive flips she attempts the switch and succeeds 10% of the time; once the switch has happened it is not undone (she wanted the biased coin):

S_i	S_{i+1}	θ
fair	fair	.9
fair	biased	.1
biased	fair	0
biased	biased	1

And the flips are governed by whichever coin is in hand:

S_i	F_i	θ
fair	h	.5
biased	h	.8

Query. The MAP over $S_1, \ldots, S_8$ given the evidence $F_1, \ldots, F_8$. The transition CPT being deterministic in the biased row, every state sequence of positive probability is fair up to some point and biased thereafter, so the query reduces to a distribution over the nine hypotheses

$$K = \min\{\, i : S_i = \text{biased} \,\} \in \{2, 3, \ldots, 8\} \cup \{\text{never}\},$$

and $\Pr(S_8 = \text{biased} \mid e)$ is the probability of a switch at all. The unnormalized weight of $K = k$ is

$$w_k = (.9)^{k-2}(.1) \cdot (.5)^{k-1} \prod_{i \geq k} \Pr(f_i \mid \text{biased}),$$

the factor $(.5)^{k-1}$ being the likelihood of the first $k-1$ flips under the fair coin; for "never", $\Pr(K = \text{never}) = (.9)^7 = .4782969$ and $w_{\text{never}} = (.9)^7(.5)^8 = .00186835$.

(a) t, t, t, h, h, h, h, h.

k	$\Pr(K = k)$	$\Pr(e \mid K = k)$	w_k	$\Pr(K = k \mid e)$
2	.1	.0065536	.00065536	.0588
3	.09	.016384	.00147456	.1322
4	.081	.04096	.00331776	.2974
5	.0729	.0256	.00186624	.1673
6	.06561	.016	.00104976	.0941
7	.059049	.01	.00059049	.0529
8	.0531441	.00625	.00033215	.0298
never	.4782969	.00390625	.00186835	.1675

The weights sum to $\Pr(e) = .011154668$. The MAP answer is $K = 4$, i.e.

$$S_1 = S_2 = S_3 = \text{fair}, \qquad S_4 = \cdots = S_8 = \text{biased},$$

at $\approx 29.74\%$, with a switch occurring at all with probability $1 - .1675 \approx 83.25\%$. The switch lands on the tails-to-heads boundary.

(b) t, t, h, h, h, h, h, h.

k	$\Pr(K = k)$	$\Pr(e \mid K = k)$	w_k	$\Pr(K = k \mid e)$
2	.1	.0262144	.00262144	.1494
3	.09	.065536	.00589824	.3362
4	.081	.04096	.00331776	.1891
5	.0729	.0256	.00186624	.1064
6	.06561	.016	.00104976	.0598
7	.059049	.01	.00059049	.0337
8	.0531441	.00625	.00033215	.0189
never	.4782969	.00390625	.00186835	.1065

The weights sum to $\Pr(e) = .017544428$. The MAP answer is $K = 3$,

$$S_1 = S_2 = \text{fair}, \qquad S_3 = \cdots = S_8 = \text{biased},$$

at $\approx 33.62\%$, with a switch occurring at all with probability $1 - .1065 \approx 89.35\%$. Again the switch sits at the tails-to-heads boundary, the longer run of heads sharpening both its location and its occurrence.

Problem 5.7. Consider the system reliability problem in Section 5.3.6 and suppose that the two fans depend on a common condition C that materializes with a probability 99.5%. In particular, as long as the condition is established, each fan will have a reliability of 90%. However, if the condition is not established, both fans will fail. Develop a Bayesian network that corresponds to this scenario and compute the overall system reliability in this case.
For reference, the reliability block diagram of Figure 5.16 runs a power supply into two parallel fans, the fans into two parallel processors, and the processors into a hard drive. The Bayesian network of Figure 5.18 has component roots E (power supply, reliability 99%), F_1, F_2 (fans, 90%), P_1, P_2 (processors, 96%), and D (hard drive, 98%), each with values avail / un avail, together with the gate variables

$$A_1 = E \wedge F_1, \quad A_2 = E \wedge F_2, \quad O_1 = A_1 \vee A_2,$$
$$A_3 = O_1 \wedge P_1, \quad A_4 = O_1 \wedge P_2, \quad O_2 = A_3 \vee A_4,$$

and $S = O_2 \wedge D$, where S is the availability of the whole system. Its reliability $\Pr(S = \text{avail})$ is $\approx 95.9\%$.

Solution. The system reliability drops to $\approx 95.42\%$. *Network:* the only change is to the fans, no longer independent roots. Add a root C with values established and not established,

C	θ_C
established	.995
not established	.005

and make C a parent of both fans, giving the edges $C \to F_1$ and $C \to F_2$. Every other node and edge of Figure 5.18 is unchanged. The fan CPTs become

C	F_i	$\theta_{F_i \mid C}$
established	avail	.90
not established	avail	0

which is the statement "given the condition each fan is 90% reliable; without it, both fans fail". The fans are now dependent, C being the only d-separator: $F_1 \not\perp F_2$ a priori while $F_1 \perp F_2 \mid C$ (Definition 4.2).
Reliability. The roots C, E, P_1, P_2, D being independent and all internal nodes deterministic gates, propagate availabilities up the block diagram, eliminating C before combining the fans:

$$\begin{aligned}\Pr(F_1 \vee F_2) &= \Pr(C)\big[1 - (1 - .9)^2\big] + \Pr(\neg C) \cdot 0 \\ &= (.995)(.99) \;=\; .98505.\end{aligned}$$

Then $O_1 = E \wedge (F_1 \vee F_2)$, since $A_1 \vee A_2 = (E \wedge F_1) \vee (E \wedge F_2)$, and E is independent of C, F_1, F_2:

$$\Pr(O_1 = \text{avail}) = (.99)(.98505) = .9751995.$$

Likewise $O_2 = O_1 \wedge (P_1 \vee P_2)$ with P_1, P_2 independent roots, so

$$\Pr(P_1 \vee P_2) = 1 - (1 - .96)^2 = 1 - .0016 = .9984,$$
$$\Pr(O_2 = \text{avail}) = (.9751995)(.9984) = .9736391808.$$

Finally $S = O_2 \wedge D$:

$$\Pr(S = \text{avail}) = (.9736391808)(.98) = .954166397184.$$

So the system reliability is $\approx 95.42\%$, down from the $\approx 95.90\%$ of Section 5.3.6. The fan block has fallen from $1 - (.1)^2 = .99$ to $(.995)(.99) = .98505$, and no amount of parallel redundancy recovers it, since $\Pr(F_1 \vee \cdots \vee F_n) = (.995)[1 - (.1)^n]$ is capped at $\Pr(C) = .995$.

4.2 Exercises 5.8–5.14

Problem 5.8. Consider the electrical network depicted in Figure 5.33 and assume that electricity can flow in either direction between two adjacent stations S_i and S_j (i.e., connected by an edge), and that electricity is flowing into the network from sources I_1 and I_2. For the network to be operational, electricity must also flow out of either outlet O_1 or outlet O_2. Describe a Bayesian network and a corresponding query that allows us to compute the reliability of this network, given the reliability r_{ij} of each connection between stations S_i and S_j. What is the reliability of the network if $r_{ij} = .99$ for all connections?

Figure 5.33 shows five stations $S_1, \ldots, S_5$ wired into a single ring, together with two sources and two outlets tapped onto that ring:

- the connections between stations are $S_1 - S_2$, $S_2 - S_3$, $S_3 - S_4$, $S_4 - S_5$ and $S_5 - S_1$ (so the five stations form a cycle);
- source I_1 feeds into the connection $S_5 - S_1$ (the arrow from I_1 enters the corner joining the wire out of S_5 to the wire into S_1);
- source I_2 feeds into the connection $S_1 - S_2$ (the arrow from I_2 enters the corner joining the wire out of S_2 to the wire into S_1);
- outlet O_1 is tapped off the connection $S_4 - S_5$;
- outlet O_2 is tapped off the connection $S_2 - S_3$.

The only unreliable elements are the five station-to-station connections; the taps themselves and the stations are perfectly reliable, and a connection either conducts over its whole length or not at all.

Solution. The network is 99.96% reliable.

The Bayesian network. The five connections of Figure 5.33 are the only unreliable elements, so introduce one root variable for each,

$$C_{12},\ C_{23},\ C_{34},\ C_{45},\ C_{51},$$

with values up and down and CPT $\Pr(C_{ij} = \mathsf{up}) = r_{ij}$; everything else is a deterministic node in the sense of Section 5.4.2, making this a "model from design" as in Section 5.3.6.

Figure 5.33 is cyclic while a Bayesian network must be a DAG, so a single variable "station S_i is energized" with its neighbours as parents is not available. Unroll the propagation in **stages** instead, as one unrolls a dynamic system into a DBN (Section 5.3.5); with five stations any station reachable from a source is reachable in at most four hops, so four stages suffice.

For $k = 0, 1, 2, 3, 4$ let E_i^k be binary, meaning "station S_i is energized by some conducting route of length at most k". Stage 0 records the injections, I_1 being tapped onto $S_5 - S_1$ and I_2 onto $S_1 - S_2$, so that a conducting tapped connection energizes both its endpoints:

$$\begin{aligned} E_1^0 &\iff C_{51} \vee C_{12}, \\ E_5^0 &\iff C_{51}, \qquad E_2^0 \iff C_{12}, \\ E_3^0 &\iff \bot, \qquad E_4^0 \iff \bot. \end{aligned}$$

One round of propagation along conducting connections gives the next stage,

$$E_i^{k+1} \iff E_i^k \ \vee \bigvee_{j \sim i} \left(E_j^k \wedge C_{ij}\right),$$

$j \sim i$ ranging over the ring neighbours of S_i. So E_i^{k+1} has parents E_i^k, the two neighbours' E_j^k and the two C_{ij}, and every edge runs from stage k to stage $k+1$ or from a root into a stage variable: the structure is acyclic. These CPTs are deterministic, written straight from the propositional equivalences in the encoding of Section 5.4.2.

Three more deterministic nodes finish the model. Outlet O_1, tapped onto $S_4 - S_5$, is fed exactly when that connection conducts and one of its endpoints is energized, and likewise O_2 on $S_2 - S_3$:

$$\begin{aligned} A_1 &\iff C_{45} \wedge (E_4^4 \vee E_5^4), \\ A_2 &\iff C_{23} \wedge (E_2^4 \vee E_3^4), \\ S &\iff A_1 \vee A_2. \end{aligned}$$

The variable S says that the network is operational.

The query. The marginal $\Pr(S = \mathsf{true})$, a probability-of-evidence query with no evidence, exactly as reliability was the marginal of the system node in Section 5.3.6; the MAP over $C_{12}, \ldots, C_{51}$ given $S = \mathsf{false}$ names the connection to blame for an outage.

Reducing the structure function. For this ring $\Pr(S)$ has a closed form, since

$$S \iff (C_{12} \wedge C_{23}) \vee (C_{45} \wedge C_{51}),$$

making $S_3 - S_4$ irrelevant. Sufficiency: C_{51} conducting lets I_1 energize S_5, and then the tap of O_1 on the conducting $S_4 - S_5$ carries current; symmetrically $C_{12} \wedge C_{23}$ feeds O_2. Necessity: all electricity originates at the two taps, so the energized stations are those reachable through conducting connections from $\{S_1, S_5\}$ (available only if C_{51} conducts) or from $\{S_1, S_2\}$ (only if C_{12} conducts). Suppose O_1 is fed, so C_{45} conducts and one of S_4, S_5 is energized.

- If C_{51} conducts, $C_{45} \wedge C_{51}$ holds.
- Otherwise all energy comes through the tap on $S_1 - S_2$, so C_{12} conducts and the energized set is what is reachable from S_2 along $S_2 - S_3 - S_4 - S_5$; reaching S_4 or S_5 then forces C_{23}, giving $C_{12} \wedge C_{23}$.

The case of O_2 follows by the symmetry of Figure 5.33: $S_2 \leftrightarrow S_5$, $S_3 \leftrightarrow S_4$ maps the ring to itself fixing S_1 and C_{34}, exchanges I_1 with I_2 and O_1 with O_2, and swaps C_{51} with C_{12} and C_{45} with C_{23}.

The number. The path sets $\{C_{12}, C_{23}\}$ and $\{C_{45}, C_{51}\}$ are disjoint and the connection variables are independent roots, so the two events are independent:

$$\Pr(S) = 1 - (1 - r_{12}r_{23})\,(1 - r_{45}r_{51}).$$

With $r_{ij} = .99$ for every connection, $r_{12}r_{23} = r_{45}r_{51} = .9801$ and

$$\begin{aligned}\Pr(S) &= 1 - (1 - .9801)^2 = 1 - (.0199)^2 \\ &= 1 - .00039601 = .99960399.\end{aligned}$$

Brute force over the 2^5 instantiations of the connection variables in the network above returns exactly 99960399/100000000, confirming the reduction.

Problem 5.9. Consider Section 5.3.6 and suppose we extend the language of RBDs to include k-out-of-n blocks. In particular, a block C is said to be of type k-out-of-n if it has n components pointing to it in the diagram, and at least k of them must be available for the subsystem represented by C to be available. Provide a systematic method for converting RBDs with k-out-of-n blocks into Bayesian networks.

Recall the semantics of Section 5.3.6: an RBD is a DAG with a single leaf, each node of which is a block. A block B represents the subsystem consisting of the component B together with all subsystems feeding into B, and that subsystem is available iff component B is available and at least one of the subsystems feeding into B is available. The conversion of Figure 5.17 gives each block a component variable, an or-node over the availabilities of the incoming subsystems, and an and-node combining the two.

Solution. Replace the or-gate of Figure 5.17 by a **threshold** gate firing when at least k of its n inputs are available, and implement that gate as a saturating counter chain rather than a single CPT of size 2^{n+1}. Figure 5.17 is then the case $k = 1$ and an and-gate the case $k = n$.

Variables. Let the RBD have blocks $B_1, \ldots, B_m$, and for a block C let $\mathrm{pa}(C) = \{D_1, \ldots, D_n\}$ be the blocks pointing to C, with threshold k, $1 \le k \le n$. Each block C gets two variables:

- a **component** variable C with values avail and $\mathsf{unavail}$;
- a **subsystem** variable S_C with values true and false, meaning "the subsystem represented by block C is available".

The component variables are roots with CPT $\Pr(C = \mathsf{avail}) = r_C$, as for the roots E, F_1, F_2, P_1, P_2, D of Figure 5.18.

Threshold gate as a counter chain. For a block C with parents $D_1, \ldots, D_n$ and threshold k, introduce counting variables

$$N_1^C, N_2^C, \ldots, N_n^C, \qquad N_j^C \in \{0, 1, \ldots, k\},$$

whose intended meaning is

$$N_j^C \;=\; \min\Big(k,\; \big|\{\, i \le j : S_{D_i} = \mathsf{true} \,\}\big|\Big),$$

the number of available incoming subsystems among the first j, saturated at k. They are chained: N_1^C has the single parent S_{D_1}, and N_j^C for $j \ge 2$ has parents N_{j-1}^C and S_{D_j}, with deterministic CPTs

$$N_1^C = \begin{cases} 1, & S_{D_1} = \mathsf{true} \\ 0, & S_{D_1} = \mathsf{false} \end{cases}$$

$$N_j^C = \begin{cases} \min(k,\, N_{j-1}^C + 1), & S_{D_j} = \mathsf{true} \\ N_{j-1}^C, & S_{D_j} = \mathsf{false}. \end{cases}$$

These realize the intended meaning by induction on j: the case $j = 1$ is immediate, and $\min(k, c_{j-1})+1$ saturates to $\min(k, c_j)$ because $\min(k, \cdot)$ is monotone and idempotent above k, while an unavailable parent leaves the count alone. (Check!) Hence at least k incoming subsystems are available iff $N_n^C = k$.

Putting a block together. Define S_C by the deterministic CPT

$$S_C = \mathsf{true} \iff \big(C = \mathsf{avail}\big) \wedge \big(N_n^C = k\big),$$

so S_C has parents C and N_n^C and a CPT of eight entries. A **source** block, one with $n = 0$, carries no threshold condition and gets

$$S_C = \mathsf{true} \iff C = \mathsf{avail},$$

as Figure 5.18 does for the power supply, the only source block of Figure 5.16. Every CPT here except those of the component roots is deterministic, written in the propositional form of Section 5.4.2.

Acyclicity and query. Every edge runs from a component root into its own S_C, from S_{D_i} into the counter chain of C, or forward along a counter chain, and the RBD is a DAG with D_i a parent of C, so the construction is a DAG. Its single leaf block L represents the whole system (Section 5.3.6), so the reliability is

$$\Pr(S_L = \mathsf{true}),$$

and the most likely explanation of a failure is the MAP over $B_1, \ldots, B_m$ given $S_L = \mathsf{false}$, as in Section 5.3.6.

Cost. A single threshold node with parents $S_{D_1}, \ldots, S_{D_n}$ would need 2^{n+1} entries, the blow-up of Section 5.4; the chain uses n nodes of at most $2(k+1)^2$ entries each, so $O(nk^2)$ numbers in all — the micro-model device that turns the noisy-or CPT of (5.2) into the network of Figure 5.30. The extremes check out: at $k = 1$ the counters are binary and the chain is a cascaded or-gate, recovering Figure 5.17, and at $k = n$ it is a cascaded and-gate.

Problem 5.10. (After Jensen). Consider a cow that may be infected with a disease that can possibly be detected by performing a milk test. The test is performed on five consecutive days, leading to five outcomes. We want to determine the state of the cow's infection over these days given the test outcomes. The prior probability of an infection on day one is 1/10,000; the test false positive rate is 5/1,000; and its false negative rate is 1/1,000. Moreover, the state of infection at a given day depends only on its state at the previous day. In particular, the probability of a new infection on a given day is 2/10,000, while the probability that an infection would persist to the next day is 7/10. (a) Describe a Bayesian network and a corresponding query that solves this problem. What is the most likely state of the cow's infection over the five days given the following test outcomes:

- (1) positive, positive, negative, positive, positive
- (2) positive, negative, negative, positive, positive
- (3) positive, negative, negative, negative, positive

(b) Assume now that the original false negative and false positive rates double on a given day in case the test has failed on the previous day. Describe a Bayesian network that captures this additional information.

Solution. **(a) The network.** A hidden Markov model, the dynamic Bayesian network of Section 5.3.5 with one slice per day. For $d = 1, \dots, 5$ take

- I_d, with values yes and no: the cow is infected on day d;
- T_d, with values $+$ and $-$: the milk test outcome on day d.

The infection state on a day depends only on the previous day's and the test outcome only on the same day's, so the edges are

$$I_1 \to I_2 \to I_3 \to I_4 \to I_5, \qquad I_d \to T_d \ \ (d = 1, \dots, 5),$$

and nothing else. The CPTs are the same in every slice:

I_1	θ_{i_1}
yes	.0001
no	.9999

I_{d-1}	I_d	$\theta_{i_d \mid i_{d-1}}$
yes	yes	.7
yes	no	.3
no	yes	.0002
no	no	.9998

I_d	T_d	$\theta_{t_d \mid i_d}$
yes	$+$	.999
yes	$-$	.001
no	$+$	.005
no	$-$	.995

the last table being fixed by the false negative rate $\Pr(T_d = - \mid I_d = \text{yes}) = 1/1{,}000$ and the false positive rate $\Pr(T_d = + \mid I_d = \text{no}) = 5/1{,}000$.

The query. The MAP over $\mathbf{M} = \{I_1, \dots, I_5\}$ given the evidence $e : T_1 = t_1, \dots, T_5 = t_5$ (Section 5.2.4). Since $\mathbf{M}$ and $\mathbf{E} = \{T_1, \dots, T_5\}$ exhaust the network, the remaining set $\mathbf{Y}$ is empty and MAP coincides exactly with MPE, the degenerate case of Exercise 5.11. As $\Pr(\mathbf{i}, e)$ factors along the chain,

$$\Pr(i_1, \dots, i_5, e) \;=\; \theta_{i_1}\theta_{t_1 \mid i_1} \prod_{d=2}^{5} \theta_{i_d \mid i_{d-1}}\theta_{t_d \mid i_d},$$

maximize by the left-to-right dynamic program (Viterbi), or enumerate the $2^5 = 32$ instantiations.

Test vector (1): $+, +, -, +, +$. The MPE is infection on all five days,

$$I_1 = I_2 = I_3 = I_4 = I_5 = \text{yes},$$

with

$$\begin{aligned} \Pr(\mathbf{i}, e) &= (10^{-4})(.999)^4(.001)(.7)^4 \\ &= 2.39141 \times 10^{-8}, \end{aligned}$$

and $\Pr(e) = 3.197439 \times 10^{-8}$, so $\Pr(\mathbf{i} \mid e) \approx 74.79\%$. The day-3 negative costs one false negative at .001, against $(.005)^4$ for the competing all-healthy story.

Test vector (2): $+, -, -, +, +$. The MPE is

$$I_1 = I_2 = I_3 = \text{no}, \qquad I_4 = I_5 = \text{yes},$$

with $\Pr(\mathbf{i}, e) = 6.912864 \times 10^{-7}$ and $\Pr(e) = 8.267956 \times 10^{-7}$, hence $\Pr(\mathbf{i} \mid e) \approx 83.61\%$: the isolated day-1 positive is a false positive, while the two consecutive positives force a new infection from day 4.

Test vector (3): $+,-,-,-,+$**.** The MPE is

$$I_1 = I_2 = I_3 = I_4 = I_5 = \mathsf{no},$$

with $\Pr(\mathbf{i}, e) = 2.460472 \times 10^{-5}$ and $\Pr(e) = 2.574235 \times 10^{-5}$, hence $\Pr(\mathbf{i} \mid e) \approx 95.58\%$: two false positives at $(.005)^2 = 2.5 \times 10^{-5}$ are cheaper than an infection that appears, disappears, and produces three false negatives.

(b) Test failures that degrade the next test. The day-d error rates now depend on whether the day-$(d-1)$ test **failed**, i.e. disagreed with the true infection state, so the model needs a variable recording that. For $d = 1, \ldots, 4$ add

$$F_d \in \{\mathsf{true}, \mathsf{false}\}, \quad \text{where}$$
$$F_d = \mathsf{true} \iff (I_d = \mathsf{yes} \wedge T_d = -) \vee (I_d = \mathsf{no} \wedge T_d = +).$$

So F_d has parents I_d, T_d and a deterministic CPT (Section 5.4.2), the new edges being

$$I_d \to F_d, \qquad T_d \to F_d, \qquad F_d \to T_{d+1},$$

so T_{d+1} has parents I_{d+1} and F_d while the chain $I_1 \to \cdots \to I_5$ is unchanged. The graph stays acyclic, every edge running within slice d or into slice $d+1$. The CPT of T_1 is the original emission table; for $d = 2, \ldots, 5$ it is indexed additionally by F_{d-1}, doubling both error rates after a failure:

I_d	F_{d-1}	T_d	$\theta_{t_d \mid i_d, f_{d-1}}$
yes	false	$+$	.999
yes	false	$-$	.001
yes	true	$+$	.998
yes	true	$-$	.002
no	false	$+$	.005
no	false	$-$	.995
no	true	$+$	.010
no	true	$-$	.990

The query is unchanged, the MAP over $I_1, \ldots, I_5$ given $T_1, \ldots, T_5$, and it can still be read off an MPE: each F_d is a deterministic function of I_d and the observed T_d, so every instantiation of the MAP variables and the evidence admits exactly one positive-probability instantiation of $\mathbf{Y} = \{F_1, \ldots, F_4\}$, which is the condition of Exercise 5.11.

Problem 5.11. Let $\mathbf{E}$ be the evidence variables in a Bayesian network, $\mathbf{M}$ be the MAP variables, and let $\mathbf{Y}$ be all other variables. Suppose that $\Pr(\mathbf{m}, \mathbf{e}) > 0$ implies that $\Pr(\mathbf{m}, \mathbf{e}, \mathbf{y}) \neq 0$ for exactly one instantiation $\mathbf{y}$. Show that the projection of an MPE solution on the MAP variables $\mathbf{M}$ is also a MAP solution.

Solution. The hypothesis makes summing out $\mathbf{Y}$ the same as maximizing it out: for every $\mathbf{m}$,

$$\Pr(\mathbf{m}, \mathbf{e}) = \sum_{\mathbf{y}} \Pr(\mathbf{m}, \mathbf{e}, \mathbf{y}) = \max_{\mathbf{y}} \Pr(\mathbf{m}, \mathbf{e}, \mathbf{y}). \tag{$*$}$$

Indeed, if $\Pr(\mathbf{m}, \mathbf{e}) > 0$ exactly one term of the sum is nonzero, so the sum equals that term, which is the maximum; and if $\Pr(\mathbf{m}, \mathbf{e}) = 0$ every term vanishes and both sides are 0.

Assume $\Pr(\mathbf{e}) > 0$ (else neither query is defined) and let $\mathbf{m}^\star \mathbf{y}^\star$ be an MPE given $\mathbf{e}$, i.e. a maximizer of $\Pr(\mathbf{m}, \mathbf{y}, \mathbf{e})$ (Section 5.2.3; the conditional and the joint differ by the positive constant $\Pr(\mathbf{e})$). For arbitrary $\mathbf{m}$, apply $(*)$, then optimality of the MPE, then $(*)$ again:

$$\begin{aligned} \Pr(\mathbf{m}, \mathbf{e}) &= \max_{\mathbf{y}} \Pr(\mathbf{m}, \mathbf{e}, \mathbf{y}) \\ &\leq \Pr(\mathbf{m}^\star, \mathbf{e}, \mathbf{y}^\star) \\ &\leq \max_{\mathbf{y}} \Pr(\mathbf{m}^\star, \mathbf{e}, \mathbf{y}) = \Pr(\mathbf{m}^\star, \mathbf{e}). \end{aligned}$$

So $\mathbf{m}^\star$ maximizes $\Pr(\mathbf{m}, \mathbf{e})$, hence also $\Pr(\mathbf{m} \mid \mathbf{e}) = \Pr(\mathbf{m}, \mathbf{e}) / \Pr(\mathbf{e})$, and is a MAP solution.

Problem 5.12. Let $\mathbf{R}$ be some root variables in a Bayesian network and let $\mathbf{Q}$ be some nonroots. Show that the probability $\Pr(\mathbf{q} \mid \mathbf{r})$ is independent of the CPTs for roots $\mathbf{R}$ for all $\mathbf{q}$ and $\mathbf{r}$ such that $\Pr(\mathbf{r}) \neq 0$.

Solution. The root parameters factor out of numerator and denominator alike and cancel. Write $\mathbf{X}$ for all variables, $\mathbf{Z} = \mathbf{X} \setminus (\mathbf{Q} \cup \mathbf{R})$ (disjoint, as $\mathbf{Q}$ are nonroots), and split the chain rule for Bayesian networks at $\mathbf{R}$: each $R \in \mathbf{R}$ is a root, so its factor is θ_r, depending on $\mathbf{r}$ alone. With

$$\kappa(\mathbf{r}) = \prod_{R \in \mathbf{R}} \theta_r, \qquad f(\mathbf{w}, \mathbf{r}) = \prod_{X \in \mathbf{X} \setminus \mathbf{R}} \theta_{x|\mathbf{u}},$$

the latter evaluated at $\mathbf{wr}$ and using no parameter of a CPT of $\mathbf{R}$, we get $\Pr(\mathbf{w}, \mathbf{r}) = \kappa(\mathbf{r}) f(\mathbf{w}, \mathbf{r})$ for every instantiation $\mathbf{w}$ of $\mathbf{X} \setminus \mathbf{R}$. Summing out,

$$\Pr(\mathbf{q}, \mathbf{r}) = \kappa(\mathbf{r}) \sum_{\mathbf{z}} f(\mathbf{qz}, \mathbf{r}),$$
$$\Pr(\mathbf{r}) = \kappa(\mathbf{r}) \sum_{\mathbf{w}} f(\mathbf{w}, \mathbf{r}).$$

Since $\Pr(\mathbf{r}) \neq 0$, both $\kappa(\mathbf{r})$ and $\sum_{\mathbf{w}} f(\mathbf{w}, \mathbf{r})$ are nonzero, so

$$\Pr(\mathbf{q} \mid \mathbf{r}) = \frac{\Pr(\mathbf{q}, \mathbf{r})}{\Pr(\mathbf{r})} = \frac{\sum_{\mathbf{z}} f(\mathbf{qz}, \mathbf{r})}{\sum_{\mathbf{w}} f(\mathbf{w}, \mathbf{r})},$$

an expression in the parameters $\theta_{x|\mathbf{u}}$, $X \notin \mathbf{R}$, only. Replacing the CPTs of $\mathbf{R}$ by any others leaves this expression unchanged, so $\Pr(\mathbf{q} \mid \mathbf{r})$ is independent of them.

Problem 5.13. Consider the two circuit models of Section 5.3.5 corresponding to two different ways of representing health variables. We showed in that section that these two models are equivalent as far as queries involving variables A, B, C, D, and E. Does this equivalence continue to hold if we extend the models to two test vectors as we did in Section 5.3.5? In particular, will the two models be equivalent with respect to queries involving variables $A, \ldots, E, A', \ldots, E'$? Explain your answer. Recall the setting. The circuit of Figure 5.14 has primary inputs A and B, an inverter X with input A and output C, an and-gate Y with inputs A, B and output D, and an or-gate Z with inputs C, D and output E. The Bayesian network has edges $A \to C$, $X \to C$, $A \to D$, $B \to D$, $Y \to D$, $C \to E$, $D \to E$, $Z \to E$. Wires take values low and high. In the first model each health variable has values ok and faulty with $\theta_{\mathsf{ok}} = .99$, $\theta_{\mathsf{faulty}} = .01$, and the output CPT is the gate function when the component is ok and uniform (.5/.5) when it is faulty. In the second model each health variable has values ok, stuckat0, stuckat1 with probabilities .99, .005, .005, and the output CPT is deterministic throughout: the gate function when ok, low when stuckat0, high when stuckat1. Extending to two test vectors as in Section 5.3.5 means adding a second copy A', B', C', D', E' of the circuit variables while keeping the same health variables X, Y, Z, which now have two children each (Figure 5.15(b)).

Solution. No: the equivalence breaks. It rested on each health variable having a **single** child, so that it could be bypassed by (5.1) of Section 5.3.4; in Figure 5.15(b) the shared health variable X has the two children C and C' (likewise Y, Z), and the models genuinely differ, because a stuck-at mode forces the **same** wrong output in both tests whereas a faulty gate with a .5/.5 output CPT reflips independently. Witness: condition on $A = A' = \mathsf{high}$ and query the inverter's outputs. Under fault modes $C = C' = \mathsf{high}$ holds exactly when $X = \mathsf{stuckat1}$ (ok or stuckat0 gives low in both tests), while under ok/faulty it needs $X = \mathsf{faulty}$ and two high flips:

$$\begin{aligned} \Pr(C = \mathsf{high}, C' = \mathsf{high} \mid A = A' = \mathsf{high}) &= .005 && \text{(fault modes)}, \\ &= (.01)(.5)(.5) = .0025 && \text{(ok/faulty)}. \end{aligned}$$

This is a query over A, C, A', C' alone, so the two models are not equivalent with respect to queries

over $A, \ldots, E, A', \ldots, E'$.

Problem 5.14. Consider the two circuit models of Section 5.3.5 corresponding to two different ways of representing health variables. We showed in that section that these two models are equivalent as far as queries involving variables A, B, C, D, and E. Show that this result generalizes to any circuit structure as long as the two models agree on the CPTs for primary inputs and the CPTs corresponding to components satisfy the following conditions:

$$\theta_{H=\mathsf{ok}} = \theta'_{H=\mathsf{ok}}$$

$$\theta_{O=0|\mathbf{i},\, H=\mathsf{faulty}} = \frac{\theta'_{H=\mathsf{stuckat0}}}{\theta'_{H=\mathsf{stuckat0}} + \theta'_{H=\mathsf{stuckat1}}}$$

$$\theta_{O=1|\mathbf{i},\, H=\mathsf{faulty}} = \frac{\theta'_{H=\mathsf{stuckat1}}}{\theta'_{H=\mathsf{stuckat0}} + \theta'_{H=\mathsf{stuckat1}}}.$$

Here H is the health of the component, O is its output, and $\mathbf{I}$ are its inputs. Moreover, θ are the network parameters when health variables have states ok and faulty, and θ' are the network parameters when health variables have states ok, stuckat0, and stuckat1.

Solution. Bypassing every health variable turns the two networks into the **same** network over the circuit variables $\mathbf{W}$ (primary inputs and wires), which gives the equivalence for every query over $\mathbf{W}$. Write Pr, θ for the ok/faulty network and Pr′, θ' for the fault-mode one; the two share the circuit part of the DAG and the primary-input CPTs.

Let f be the Boolean function of a given component, so that the CPTs of its output are

$$\begin{aligned}
\theta_{O=o|\mathbf{i},\,\mathsf{ok}} &= \theta'_{O=o|\mathbf{i},\,\mathsf{ok}} = [\, o = f(\mathbf{i}) \,], \\
\theta'_{O=o|\mathbf{i},\,\mathsf{stuckat0}} &= [\, o = 0 \,], \\
\theta'_{O=o|\mathbf{i},\,\mathsf{stuckat1}} &= [\, o = 1 \,],
\end{aligned}$$

where $[\,\cdot\,]$ is 1 when the enclosed statement holds and 0 otherwise; the stuck-at CPTs are deterministic because a stuck-at gate ignores its inputs. The exercise's conditions do not mention $\mathbf{i}$, so they force $\theta_{O=o|\mathbf{i},\mathsf{faulty}}$ to be one number $\theta_{O=o|\mathsf{faulty}}$ for all $\mathbf{i}$.

Each health variable H is a root whose only child is its own component's output O, and H is neither query nor evidence here, so Section 5.3.4 permits bypassing it: delete H and reset

$$\theta^{\text{new}}_{o|\mathbf{i}} = \sum_h \theta_{o|\mathbf{i},h}\, \theta_h$$

by (5.1). The bypass adds no edges and touches no other CPT, so the remaining health variables still have one child each and all of them may be bypassed in turn, in either network; what is left in both cases is the circuit DAG on $\mathbf{W}$.

The bypassed wire CPTs then coincide. Abbreviate

$$p = \theta_{H=\mathsf{ok}}, \quad s_0 = \theta'_{H=\mathsf{stuckat0}}, \quad s_1 = \theta'_{H=\mathsf{stuckat1}}.$$

Since $\theta_{H=\mathsf{faulty}} = 1 - p$, $\theta'_{H=\mathsf{ok}} = 1 - s_0 - s_1$, and the stuck-at CPTs are deterministic,

$$\begin{aligned}
\theta^{\text{new}}_{o|\mathbf{i}} &= [\, o = f(\mathbf{i}) \,]\, p + \theta_{O=o|\mathsf{faulty}}(1-p), \\
\theta'^{\text{new}}_{o|\mathbf{i}} &= [\, o = f(\mathbf{i}) \,]\, \theta'_{H=\mathsf{ok}} + [\, o = 0 \,]\, s_0 + [\, o = 1 \,]\, s_1.
\end{aligned}$$

The first condition gives $\theta'_{H=\mathsf{ok}} = p$, hence $1-p = s_0 + s_1$; the second and third then give $\theta_{O=o|\mathsf{faulty}}(1-p) = s_o$ for $o = 0$ and $o = 1$. (If $s_0 + s_1 = 0$ those ratios are undefined, but then the component never fails in either model and that identity reads $0 = 0$.) Substituting,

$$\theta^{\text{new}}_{o|\mathbf{i}} = [\, o = f(\mathbf{i}) \,]\, p + s_o = \theta'^{\text{new}}_{o|\mathbf{i}}$$

for every $\mathbf{i}$ and o. So after the bypasses the two networks have the same DAG, the same primary-input CPTs (assumed) and the same wire CPTs: they are the same network. Since bypassing preserves $\Pr(\mathbf{q}, \mathbf{e})$ over the surviving variables (Section 5.3.4),

$$\Pr(\mathbf{q}, \mathbf{e}) = \Pr'(\mathbf{q}, \mathbf{e}) \qquad \text{for all } \mathbf{Q}, \mathbf{E} \subseteq \mathbf{W},$$

for any feedback-free assembly of components, as required.

4.3 Exercises 5.15–5.16

Problem 5.15. Consider Exercise 5.14. Recall the setting of Section 5.3.5: a combinational digital circuit is modeled by a Bayesian network whose variables are the circuit wires together with one health variable H per component. The primary inputs and all health variables are roots; the output O of a component has as parents the component inputs I and the component health H. Two parametrizations of this same structure are considered. In the first, each health variable has the values ok and faulty; it induces the distribution Pr with parameters θ. In the second, each health variable has the values ok, stuckat0, and stuckat1; it induces the distribution Pr′ with parameters θ'. When a component is healthy its output is given by the gate function in both models, and the two models agree on the CPTs of the primary inputs. When a component is stuck at 0 (respectively 1) its output is 0 (respectively 1) with probability one, irrespective of its inputs, where the wire values 0 and 1 are the low and high of Section 5.3.5. Exercise 5.14 assumes that the remaining component parameters are related by

$$\begin{aligned}
\theta_{H=\text{ok}} &= \theta'_{H'=\text{ok}}, \\
\theta_{O=0|i,\, H=\text{faulty}} &= \frac{\theta'_{H'=\text{stuckat0}}}{\theta'_{H'=\text{stuckat0}} + \theta'_{H'=\text{stuckat1}}}, \\
\theta_{O=1|i,\, H=\text{faulty}} &= \frac{\theta'_{H'=\text{stuckat1}}}{\theta'_{H'=\text{stuckat0}} + \theta'_{H'=\text{stuckat1}}}.
\end{aligned}$$

Let $\mathbf{H}$ and $\mathbf{H}'$ be the health variables in the corresponding networks and let $\mathbf{X}$ be all other variables in either network (that is, the circuit wires, which are common to both networks). Let h be an instantiation of variables $\mathbf{H}$, assigning either ok or faulty to each variable in $\mathbf{H}$. Let h' be defined as follows:

$$\begin{aligned}
h' \stackrel{\text{def}}{=} &\Big(\textstyle\bigwedge_{h \models H=\text{ok}} (H' = \text{ok}) \Big) \\
&\wedge \Big(\textstyle\bigwedge_{h \models H=\text{faulty}} \big((H' = \text{stuckat0}) \vee (H' = \text{stuckat1}) \big) \Big).
\end{aligned}$$

Show that $\Pr(x, h) = \Pr'(x, h')$ for all x and h, where Pr is the distribution induced by the network with health states ok/faulty and Pr′ is the distribution induced by the network with health states ok/stuckat0/stuckat1.

Solution. Both sides factor componentwise, and the identity reduces to one local check per component. Index the components $c = 1, \ldots, n$; component c has health H_c (resp. H'_c), inputs $\mathbf{I}_c$ and output O_c, and x assigns i_c, o_c to these wires and a to each primary input $A \in \mathbf{R}$. By the chain rule (4.2),

$$\Pr(x, h) = \pi(x) \prod_{c=1}^{n} \theta_{h_c}\, \theta_{o_c|i_c,\, h_c}, \qquad \Pr'(x, h'') = \pi(x) \prod_{c=1}^{n} \theta'_{h''_c}\, \theta'_{o_c|i_c,\, h''_c},$$

with the common primary-input factor $\pi(x) = \prod_{A \in \mathbf{R}} \theta_a = \prod_{A \in \mathbf{R}} \theta'_a$, the two models being assumed to agree there.

Reading off its definition, an instantiation h'' of $\mathbf{H}'$ satisfies h' exactly when $h''_c \in \mathcal{V}_c(h)$ for every c, where $\mathcal{V}_c(h) = \{\text{ok}\}$ if $h \models H_c = \text{ok}$ and $\mathcal{V}_c(h) = \{\text{stuckat0}, \text{stuckat1}\}$ if $h \models H_c = \text{faulty}$. So h' is the disjoint union of the instantiations in the Cartesian product $\mathcal{V}_1(h) \times \cdots \times \mathcal{V}_n(h)$, and additivity

followed by the distributive law gives

$$\Pr'(x,h') \;=\; \sum_{h''\models h'} \Pr'(x,h'') \;=\; \pi(x)\prod_{c=1}^{n}\Big(\sum_{v\,\in\,\mathcal{V}_c(h)} \theta'_{H'_c=v}\,\theta'_{o_c|i_c,\,v}\Big).$$

It therefore suffices to prove, componentwise, the local identity

$$\theta_{h_c}\,\theta_{o_c|i_c,\,h_c} \;=\; \sum_{v\,\in\,\mathcal{V}_c(h)} \theta'_{H'_c=v}\,\theta'_{o_c|i_c,\,v}. \qquad (\star)$$

(i) $h \models H_c = \mathit{ok}$. The sum is the single term $\theta'_{H'_c=\mathrm{ok}}\,\theta'_{o_c|i_c,\,\mathrm{ok}}$, whose factors are $\theta_{H_c=\mathrm{ok}}$ by the first condition of Exercise 5.14 and $\theta_{o_c|i_c,\,\mathrm{ok}}$ since both models use the same gate function when healthy.
(ii) $h \models H_c = \mathit{faulty}$. Put $\alpha = \theta'_{H'_c=\mathrm{stuckat0}}$, $\beta = \theta'_{H'_c=\mathrm{stuckat1}}$, so that the first condition gives $\theta_{H_c=\mathrm{faulty}} = 1-\theta'_{H'_c=\mathrm{ok}} = \alpha+\beta$. The stuck-at CPTs are deterministic and input-independent, so exactly one term of the two-term sum survives, namely α when $o_c = 0$ and β when $o_c = 1$; the second and third conditions give the same two values on the left, since $\theta_{H_c=\mathrm{faulty}}\,\theta_{o_c|i_c,\,\mathrm{faulty}}$ is $(\alpha+\beta)\alpha/(\alpha+\beta)$ or $(\alpha+\beta)\beta/(\alpha+\beta)$. (If $\alpha+\beta = 0$ the ratios are undefined, but then also $\theta_{H_c=\mathrm{faulty}} = 0$ and both sides of $(\star)$ vanish.) Note this is where the input-independence of $\theta_{O=0|i,H=\mathrm{faulty}}$ is needed.
Substituting $(\star)$ for every c yields $\Pr'(x,h') = \Pr(x,h)$ for all x and h.

Problem 5.16. Consider a DAG G with one root node S and one leaf node T, where every edge $U \to X$ represents a communication link between U and X – a link is up if U can communicate with X and down otherwise. In such a model, nodes S and T can communicate iff there exists a directed path from S to T where all links are up. Suppose that each edge $U \to X$ is labeled with a probability p representing the reliability of communication between U and X. Describe a Bayesian network and a corresponding query that computes the reliability of communication between S and T. The Bayesian network should have a size that is proportional to the size of the DAG G.

Solution. The reliability is the prior marginal $\Pr(R_T = \mathrm{true})$ in the following network, where R_X means "some directed S-to-X path has all links up." Write $G = (V,E)$, let p_{UX} label the edge $U \to X$, and take the link states mutually independent, as intended.

- For each edge $U \to X$, a root L_{UX} with values up, down and $\theta_{L_{UX}=\mathrm{up}} = p_{UX}$. These are the only stochastic variables.
- R_S, a root with $\theta_{R_S=\mathrm{true}} = 1$ (the empty path reaches S).
- For each non-root X with parents $U_1,\dots,U_k$ in G, a cascade of deterministic Boolean variables $O_{X,1},\dots,O_{X,k}$, and $R_X := O_{X,k}$:

$$\begin{aligned} O_{X,1} \;&=\; R_{U_1} \wedge (L_{U_1X} = \mathrm{up}),\\ O_{X,j} \;&=\; O_{X,j-1} \;\vee\; \big(R_{U_j} \wedge (L_{U_jX} = \mathrm{up})\big), \qquad j = 2,\dots,k. \end{aligned}$$

The cascade is the micro-model device of Section 5.4.1: the direct CPT $R_X = \bigvee_j (R_{U_j} \wedge (L_{U_jX} = \mathrm{up}))$ would have 2^{2k+1} entries, whereas each $O_{X,j}$ has at most three parents and so at most 16. Every node other than S has a parent (S is the only root), so no cascade is empty. The network has $2|E|+1$ variables and $O(|E|)$ entries, proportional to G; it is acyclic, since ordering the link roots and R_S first and then the O-blocks by a topological order of G, internally by j, is a topological order of it.
Correctness is by induction along that order: the cascade unwinds to $R_X(\ell) = \bigvee_j (R_{U_j}(\ell) \wedge (\ell_{U_jX} = \mathrm{up}))$ for each joint link state ℓ, and an all-up S-to-X path is exactly an all-up S-to-U_j path followed by an up edge $U_j \to X$, which by hypothesis holds iff $R_{U_j}(\ell) = \mathrm{true}$; the base case $X = S$ is the empty path. Since the L_{UX} are independent roots and every other variable is a function of ℓ,

$$\Pr(R_T = \mathrm{true}) \;=\; \sum_{\ell} [\,R_T(\ell) = \mathrm{true}\,] \prod_{U\to X\,\in\,E} \Pr(\ell_{UX}) \;=\; \mathrm{rel}(S,T).$$

5 Inference by Variable Elimination

Contents

5.1 Exercises 6.1–6.7 69
5.2 Exercises 6.8–6.14 74
5.3 Exercises 6.15–6.18 78

5.1 Exercises 6.1–6.7

Problem 6.1. Consider the Bayesian network in Figure 6.1 on Page 127. Its structure is

$$A \to B, \quad A \to C,$$
$$B \to D, \quad C \to D, \quad C \to E,$$

where A is Winter?, B is Sprinkler?, C is Rain?, D is Wet Grass?, and E is Slippery Road?. All variables are binary with values true/false. The CPTs are:

A	Θ_A
true	.6
false	.4

A	B	$\Theta_{B\|A}$
true	true	.2
true	false	.8
false	true	.75
false	false	.25

A	C	$\Theta_{C\|A}$
true	true	.8
true	false	.2
false	true	.1
false	false	.9

B	C	D	$\Theta_{D\|BC}$
true	true	true	.95
true	true	false	.05
true	false	true	.9
true	false	false	.1
false	true	true	.8
false	true	false	.2
false	false	true	0
false	false	false	1

C	E	$\Theta_{E\|C}$
true	true	.7
true	false	.3
false	true	0
false	false	1

(a) Use variable elimination to compute the marginals $\Pr(Q, e)$ and $\Pr(Q \mid e)$ where $Q = \{E\}$ and $e : D = \text{false}$. Use the min-degree heuristic for determining the elimination order, breaking ties by choosing variables that come first in the alphabet. Show all steps.
(b) Prune the network given the query $Q = \{A\}$ and evidence $e : E = \text{false}$. Show the steps of node and edge pruning separately.

Solution. Part (a). $\Pr(E, D = \text{false}) = (.05957, .24093)$ and $\Pr(E \mid D = \text{false}) \approx (.1982, .8018)$, obtained as follows.
Pruning (Algorithm 8, VE_{PR}) does nothing here: $Q \cup \mathbf{E} = \{E, D\}$ contains both leaves D and E, and the evidence variable D is childless.
The interaction graph (Definition 6.4) of the five CPTs is

$$G: \quad A - B, \;\; A - C, \;\; B - C, \;\; B - D, \;\; C - D, \;\; C - E,$$

with degrees $2, 3, 4, 2, 1$ for $A, \ldots, E$. Min-degree over the non-query variables, alphabetical tie-break, gives $\pi = A, B, D, C$:

- A and D tie at degree 2, take A; B, C already adjacent, leaving $B - C,\ B - D,\ C - D,\ C - E$.
- B and D tie at degree 2, take B; C, D already adjacent, leaving $C - D,\ C - E$.
- $d(D) = 1 < d(C) = 2$, take D, leaving $C - E$; then C.

The constructed factors are over $\{B, C\}, \{C, D\}, \{C\}, \{E\}$, so the width is 2. Reducing on $e : D = \text{false}$ (Definition 6.5) touches only $\Theta_{D|BC}$, whose surviving rows are

B	C	D	$\Theta^e_{D\|BC}$
true	true	false	.05
true	false	false	.1
false	true	false	.2
false	false	false	1

Eliminating A from $\Theta_A, \Theta_{B|A}, \Theta_{C|A}$ gives $f_1(B, C) = \sum_A \Theta_A \, \Theta_{B|A} \, \Theta_{C|A}$, e.g. $f_1(\text{true}, \text{true}) = (.6)(.2)(.8) + (.4)(.75)(.1) = .126$:

B	C	f_1
true	true	.126
true	false	.294
false	true	.394
false	false	.186

Eliminating B from f_1 and $\Theta^e_{D|BC}$, then D from the result, then C against $\Theta_{E|C}$:

$$
\begin{aligned}
f_2(\text{true}, \text{false}) &= (.126)(.05) + (.394)(.2) = .0851, \\
f_2(\text{false}, \text{false}) &= (.294)(.1) + (.186)(1) = .2154, \qquad f_2(C, \text{true}) = 0, \\
f_3(C) &= \textstyle\sum_D f_2(C, D) = (.0851, .2154), \\
f_4(\text{true}) &= (.0851)(.7) + (.2154)(0) = .05957, \\
f_4(\text{false}) &= (.0851)(.3) + (.2154)(1) = .24093.
\end{aligned}
$$

Thus $f_4 = \Pr(E, D = \text{false})$, and $\Pr(D = \text{false}) = .05957 + .24093 = .3005$, so dividing gives $.05957/.3005 \approx .1982$ and $.24093/.3005 \approx .8018$.

Part (b). pruneNetwork$(\mathcal{N}, \{A\}, e) = A \to C \to E$, with CPTs Θ_A, $\Theta_{C|A}$, $\Theta_{E|C}$.

Node pruning (Theorem 6.4). Delete leaves outside $Q \cup \mathbf{E} = \{A, E\}$: first D with $\Theta_{D|BC}$, then the newly exposed leaf B with $\Theta_{B|A}$. The only leaf left is $E \in Q \cup \mathbf{E}$ (C is not a leaf, as $C \to E$ remains), so this halts.

Edge pruning (Theorem 6.5). No edge leaves an evidence variable, the sole one being the leaf E, so no edge is deleted and no CPT reduced.

Problem 6.2. Consider the Bayesian network in Figure 6.10. Its structure is the tree

$$A \to B, \qquad B \to C, \qquad B \to D,$$

with all variables binary (true/false) and CPTs

A	Θ_A
true	.2
false	.8

A	B	$\Theta_{B\|A}$
true	true	.5
true	false	.5
false	true	.0
false	false	1.0

B	C	$\Theta_{C\|B}$
true	true	1.0
true	false	.0
false	true	.5
false	false	.5

B	D	$\Theta_{D\mid B}$
true	true	.75
true	false	.25
false	true	.3
false	false	.7

Use variable elimination with ordering D, C, A to compute $\Pr(B, C = \text{true})$, $\Pr(C = \text{true})$, and $\Pr(B \mid C = \text{true})$.

Solution. $\Pr(B, C = \text{true}) = (.1, .45)$, $\Pr(C = \text{true}) = .55$, and $\Pr(B \mid C = \text{true}) = (2/11, 9/11) \approx (.1818, .8182)$.
Running VE_{PR2} (Algorithm 7), reduction on $e : C = \text{true}$ (Definition 6.5) touches only $\Theta_{C\mid B}$, leaving the rows

B	C	$\Theta^{e}_{C\mid B}$
true	true	1.0
false	true	.5

Eliminating $\pi = D, C, A$ in turn:

$f_1(B) = \sum_D \Theta_{D\mid B} = 1$ (a CPT sums to 1 over its head),
$f_2(B) = \sum_C \Theta^{e}_{C\mid B} = (1.0,\ .5)$ (the likelihood of e given b),
$f_3(\text{true}) = (.2)(.5) + (.8)(.0) = .1, \qquad f_3(\text{false}) = (.2)(.5) + (.8)(1.0) = .9.$

Line 7 returns the product of the surviving factors, all over B:

$$\Pr(B = \text{true},\ C = \text{true}) = (1)(1.0)(.1) = .1,$$
$$\Pr(B = \text{false},\ C = \text{true}) = (1)(.5)(.9) = .45,$$

whence $\Pr(C = \text{true}) = .1 + .45 = .55$ and, by Bayes conditioning, $.1/.55 = 2/11$ and $.45/.55 = 9/11$.

Problem 6.3. Consider a chain network $C_0 \to C_1 \to \cdots \to C_n$. Suppose that variable C_t, for $t \geq 0$, denotes the health state of a component at time t. In particular, let each C_t take on states ok and faulty. Let C_0 denote component birth where $\Pr(C_0 = \text{ok}) = 1$ and $\Pr(C_0 = \text{faulty}) = 0$. For each $t > 0$, let the CPT of C_t be

$$\Pr(C_t = \text{ok} \mid C_{t-1} = \text{ok}) = \lambda,$$
$$\Pr(C_t = \text{faulty} \mid C_{t-1} = \text{faulty}) = 1.$$

That is, if a component is healthy at time $t-1$, then it remains healthy at time t with probability λ. If a component is faulty at time $t-1$, then it remains faulty at time t with probability 1.
(a) Using variable elimination with variable ordering C_0, C_1, compute $\Pr(C_2)$.
(b) Using variable elimination with variable ordering $C_0, C_1, \ldots, C_{n-1}$, compute $\Pr(C_n)$.

Solution. $\Pr(C_2) = (\lambda^2,\ 1 - \lambda^2)$ and $\Pr(C_n) = (\lambda^n,\ 1 - \lambda^n)$ over $(\text{ok}, \text{faulty})$. The CPTs, completed by the requirement that each row sums to 1, are

C_0	Θ_{C_0}
ok	1
faulty	0

and, for every $t > 0$,

C_{t-1}	C_t	$\Theta_{C_t\mid C_{t-1}}$
ok	ok	λ
ok	faulty	$1 - \lambda$
faulty	ok	0
faulty	faulty	1

There is no evidence, and the query variable is the chain's only leaf, so nothing is reduced or pruned.
Part (a). Eliminating C_0, then C_1:

$$f_1(\text{ok}) = (1)(\lambda) + (0)(0) = \lambda, \qquad f_1(\text{faulty}) = (1)(1-\lambda) + (0)(1) = 1-\lambda,$$
$$f_2(\text{ok}) = \lambda \cdot \lambda + (1-\lambda)\cdot 0 = \lambda^2, \quad f_2(\text{faulty}) = \lambda(1-\lambda) + (1-\lambda) = 1-\lambda^2.$$

Part (b). By induction, eliminating $C_0, \ldots, C_{t-1}$ leaves the single factor $f_t(\text{ok}) = \lambda^t$, $f_t(\text{faulty}) = 1-\lambda^t$: the case $t = 1$ is above, and since C_t occurs only in f_t and $\Theta_{C_{t+1}|C_t}$,

$$f_{t+1}(\text{ok}) = \lambda^t \cdot \lambda + (1-\lambda^t)\cdot 0 = \lambda^{t+1},$$
$$f_{t+1}(\text{faulty}) = \lambda^t(1-\lambda) + (1-\lambda^t)\cdot 1 = 1-\lambda^{t+1}.$$

At $t = n$ the order has covered every non-query variable, so f_n is the marginal $\Pr(C_n)$.

Problem 6.4. Prove the technique of bypassing nodes as given by Equation 5.1 on Page 91. That is, let $\mathcal{N}$ be a Bayesian network containing a variable X that is neither a query variable nor an evidence variable, and suppose X has a single child Y. Let $\mathbf{U}$ be the parents of X and let $\mathbf{V}$ be the parents of Y other than X (see Figure 5.13). Bypassing X means deleting X from the network, making $\mathbf{U}$ parents of Y, and replacing the CPT of Y by the CPT over the new family $Y\mathbf{UV}$ defined by

$$\theta'_{y|\mathbf{uv}} = \sum_x \theta_{y|x\mathbf{v}}\, \theta_{x|\mathbf{u}}. \qquad (5.1)$$

Show that the resulting network $\mathcal{N}'$ is a well-formed Bayesian network and that its distribution $\Pr'$ satisfies $\Pr(\mathbf{q}, \mathbf{e}) = \Pr'(\mathbf{q}, \mathbf{e})$ for all instantiations of the query variables $\mathbf{Q}$ and evidence variables $\mathbf{E}$, neither of which contains X.

Solution. Bypassing is one step of variable elimination on X, whose output happens to be a legitimate CPT. Write $\mathbf{Z}$ for the variables of $\mathcal{N}$ and $\mathbf{Z}' = \mathbf{Z} \setminus \{X\}$ for those of $\mathcal{N}'$.
Well-formedness. $\mathcal{N}'$ is acyclic: all new edges end at Y, so a cycle in $\mathcal{N}'$ could use only one of them, and replacing that $U \to Y$ by the path $U \to X \to Y$ of $\mathcal{N}$ would produce a cycle in $\mathcal{N}$. The new table is a CPT: its entries are nonnegative and, for every $\mathbf{uv}$,

$$\sum_y \theta'_{y|\mathbf{uv}} = \sum_x \theta_{x|\mathbf{u}} \sum_y \theta_{y|x\mathbf{v}} = \sum_x \theta_{x|\mathbf{u}} \cdot 1 = 1.$$

$\Pr'$ *is* $\Pr$ *with* X *summed out.* Since Y is X's only child, every $W \notin \{X, Y\}$ keeps its family and CPT. Fix $\mathbf{z}'$ and let $y, \mathbf{u}, \mathbf{v}, \mathbf{p}_W$ be the values it assigns. By the chain rule in $\mathcal{N}'$, then (5.1), then Theorem 6.1 (the product over $W \neq X, Y$ does not mention X, so it moves inside the sum), then the chain rule in $\mathcal{N}$:

$$\Pr'(\mathbf{z}') = \theta'_{y|\mathbf{uv}} \prod_{W \neq X,Y} \theta_{w|\mathbf{p}_W} = \Big(\sum_x \theta_{y|x\mathbf{v}}\, \theta_{x|\mathbf{u}}\Big) \prod_{W \neq X,Y} \theta_{w|\mathbf{p}_W}$$
$$= \sum_x \Big(\theta_{x|\mathbf{u}}\, \theta_{y|x\mathbf{v}} \prod_{W \neq X,Y} \theta_{w|\mathbf{p}_W}\Big) = \sum_x \Pr(\mathbf{z}', x) = \Pr(\mathbf{z}').$$

Queries. For $\mathbf{Q} \cup \mathbf{E} \subseteq \mathbf{Z}'$, the instantiations $\mathbf{z}$ of $\mathbf{Z}$ compatible with $\mathbf{qe}$ are exactly the pairs $(\mathbf{z}', x)$ with $\mathbf{z}'$ compatible, so

$$\Pr'(\mathbf{q}, \mathbf{e}) = \sum_{\mathbf{z}' \sim \mathbf{qe}} \sum_x \Pr(\mathbf{z}', x) = \sum_{\mathbf{z} \sim \mathbf{qe}} \Pr(\mathbf{z}) = \Pr(\mathbf{q}, \mathbf{e}).$$

Problem 6.5. Consider a naive Bayes structure with edges $X \to Y_1, \ldots, X \to Y_n$.
(a) What is the width of variable order $Y_1, \ldots, Y_n, X$?

(b) What is the width of variable order $X, Y_1, \ldots, Y_n$?

Solution. (a) Width 1. (b) Width n.
The interaction graph (Definition 6.4) of Θ_X and the $\Theta_{Y_i|X}$ is the star with centre X:

$$G: \quad X - Y_1, \quad X - Y_2, \quad \ldots, \quad X - Y_n,$$

with no edge between any Y_i, Y_j. Eliminating a variable builds a factor over its current neighbours, then connects them pairwise and deletes it (Section 6.6).
(a) Each Y_i has the single neighbour X, so its elimination builds a factor over $\{X\}$, adds no fill-in edge, and leaves a smaller star; X is eliminated last from an isolated node, giving a factor over $\emptyset$. The sizes are $1, \ldots, 1, 0$.
(b) X has all n neighbours $Y_1, \ldots, Y_n$, so eliminating it first builds

$$f_1(Y_1, \ldots, Y_n) \;=\; \sum_X \Theta_X \prod_{i=1}^{n} \Theta_{Y_i|X}$$

over n variables and fills in every pair Y_iY_j, leaving the complete graph on $Y_1, \ldots, Y_n$. Eliminating from a complete graph keeps it complete, so Y_i then yields a factor of size $n - i$; the maximum is n.

Problem 6.6. Consider a two-layer Bayesian network $\mathcal{N}$ with nodes $\mathbf{X} \cup \mathbf{Y}$, $\mathbf{X} \cap \mathbf{Y} = \emptyset$, where each $X \in \mathbf{X}$ is a root node and each $Y \in \mathbf{Y}$ is a leaf node with at most k parents. In such a network, all edges in $\mathcal{N}$ are directed from a node in $\mathbf{X}$ to a node in $\mathbf{Y}$. Suppose we are interested in computing $\Pr(Y)$ for all $Y \in \mathbf{Y}$. What is the effective treewidth for $\mathcal{N}$ for each one of these queries?

Solution. The effective treewidth for the query $\Pr(Y)$ is $|\mathbf{U}_Y| \leq k$, the number of parents of Y, whatever the treewidth of $\mathcal{N}$ itself.
With $\mathbf{E} = \emptyset$, pruneEdges deletes nothing, and node pruning (Theorem 6.4) first deletes every leaf $Y' \in \mathbf{Y} \setminus \{Y\}$, then every root of $\mathbf{X} \setminus \mathbf{U}_Y$, which has just become childless. It halts there: each $U \in \mathbf{U}_Y$ still has the child Y, and $Y \in Q$. So pruneNetwork$(\mathcal{N}, \{Y\}, \emptyset)$ is Y with its parents $\mathbf{U}_Y$.
Its factor $\Theta_{Y|\mathbf{U}_Y}$ mentions all $|\mathbf{U}_Y| + 1$ surviving variables, so the interaction graph is complete on them, and every elimination order of a complete graph on m nodes has width $m - 1$ (the first variable eliminated is adjacent to the other $m - 1$, and eliminating leaves the graph complete). Hence by Definition 6.6 the effective treewidth is $|\mathbf{U}_Y|$.

Problem 6.7. Prune the network in Figure 6.11 given the following queries. The network structure of Figure 6.11 has nodes A, B, C, D, E, F, G, H and edges

$$\begin{aligned} A \to B, &\quad A \to C, \\ B \to D, &\quad C \to E, \\ D \to F, &\quad E \to G, \\ F \to H, &\quad G \to H. \end{aligned}$$

That is, A is the unique root, the two chains $B \to D \to F$ and $C \to E \to G$ descend from it, and they meet again at the unique leaf H.
(a) $Q = \{B, E\}$ and $e : A = \text{true}$
(b) $Q = \{A\}$ and $e : B = \text{true}, F = \text{true}$
Show the steps of node and edge pruning separately.

Solution. Node pruning iteratively deletes leaves outside $Q \cup \mathbf{E}$ with their CPTs (Theorem 6.4); edge pruning deletes every edge out of an evidence variable and reduces the child's CPT at the observed value (Theorem 6.5).
Part (a): $Q \cup \mathbf{E} = \{A, B, E\}$.

Node pruning.

iteration	leaf nodes	action
1	H	$H \notin \{A, B, E\}$: delete H, $\Theta_{H\|FG}$
2	F, G	delete both, with $\Theta_{F\|D}$ and $\Theta_{G\|E}$
3	D, E	delete D and $\Theta_{D\|B}$; keep E (it is in Q)
4	B, E	both are in Q: stop

(C is never a leaf, as $C \to E$ survives.) This leaves $A \to B$, $A \to C \to E$ with CPTs Θ_A, $\Theta_{B|A}$, $\Theta_{C|A}$, $\Theta_{E|C}$.
Edge pruning. The evidence variable A has the surviving outgoing edges $A \to B$ and $A \to C$; delete both and set

$$\Theta_B := \Theta_{B|A}^{A=\text{true}}, \qquad \Theta_C := \Theta_{C|A}^{A=\text{true}},$$

each a one-variable table. The pruned network is A and B isolated together with $C \to E$, with CPTs Θ_A, Θ_B, Θ_C, $\Theta_{E|C}$; a further node-pruning round removes nothing.
Part (b): $Q \cup \mathbf{E} = \{A, B, F\}$.
Node pruning.

iteration	leaf nodes	action
1	H	delete H and $\Theta_{H\|FG}$
2	F, G	keep F (in $\mathbf{E}$); delete G, $\Theta_{G\|E}$
3	E, F	delete E and $\Theta_{E\|C}$; keep F
4	C, F	delete C and $\Theta_{C\|A}$; keep F
5	F	$F \in \mathbf{E}$: stop

This leaves the chain $A \to B \to D \to F$ with CPTs Θ_A, $\Theta_{B|A}$, $\Theta_{D|B}$, $\Theta_{F|D}$.
Edge pruning. F is now childless, so only $B \to D$ goes; replace $\Theta_{D|B}$ by $\Theta_D := \Theta_{D|B}^{B=\text{true}}$. The pruned network is the two disconnected edges $A \to B$ and $D \to F$, with CPTs Θ_A, $\Theta_{B|A}$, Θ_D, $\Theta_{F|D}$; a further node-pruning round removes nothing, both leaves being in $\mathbf{E}$.

5.2 Exercises 6.8–6.14

Problem 6.8. What is the width of the order A, B, C, D, E, F, G, H with respect to the network in Figure 6.11?
Figure 6.11 is the Bayesian network structure on the eight variables A, B, C, D, E, F, G, H with the following edges:

$$A \to B, \quad A \to C, \quad B \to D, \quad C \to E,$$
$$D \to F, \quad E \to G, \quad F \to H, \quad G \to H.$$

That is, the DAG splits at the root A into two chains $A \to B \to D \to F$ and $A \to C \to E \to G$, which rejoin at the single sink H, whose parents are F and G.

Solution. Width 2. The interaction graph of the CPTs (Definition 6.4) is the moral graph, since the CPT of X is a factor over $\{X\} \cup \mathbf{U}$ and a factor's variables form a clique; only H has two parents, so moralization adds just $F - G$:

$$A - B,\ A - C,\ B - D,\ C - E,\ D - F,\ E - G,\ F - H,\ G - H,\ F - G.$$

Running OrderWidth (Algorithm 4), recording the degree d of the eliminated variable and connecting its neighbours pairwise:

step	eliminated	neighbours in current graph	d	fill-in edges added
1	A	B, C	2	$B - C$
2	B	C, D	2	$C - D$
3	C	D, E	2	$D - E$
4	D	E, F	2	$E - F$
5	E	F, G	2	none ($F - G$ present)
6	F	G, H	2	none ($G - H$ present)
7	G	H	1	none
8	H	none	0	none

The fill-in edge simply walks the cut down the two chains – after step 4 the graph is $E - F, E - G, F - G, F - H, G - H$, so E needs no new edge and what is left is the triangle F, G, H. Hence $\max_i d_i = 2$.

Problem 6.9. Compute an elimination order for the variables in Figure 6.11 using the min-degree method. In case of a tie, choose variables that come first alphabetically.
Figure 6.11 is the Bayesian network structure on the variables A, B, C, D, E, F, G, H with edges

$$A \to B, \quad A \to C, \quad B \to D, \quad C \to E,$$
$$D \to F, \quad E \to G, \quad F \to H, \quad G \to H,$$

that is, two chains $A \to B \to D \to F$ and $A \to C \to E \to G$ leaving the root A and rejoining at the sink H, whose parents are F and G.

Solution. Min-degree with alphabetical tie-breaking returns $\pi = A, B, C, D, E, F, G, H$, of width 2.
Algorithm 5 runs on the moral graph, which adds only $F - G$ (the sole node with two parents is H):

$$A - B,\ A - C,\ B - D,\ C - E,\ D - F,\ E - G,\ F - H,\ G - H,\ F - G,$$

with degrees

node	A	B	C	D	E	F	G	H
degree	2	2	2	2	2	3	3	2

Every node but F, G starts at degree 2, so the first pick is A; eliminating it (neighbours B, C, fill-in $B - C$) leaves B at degree 2, and the same step repeats along the chain: B (fill-in $C - D$), C ($D - E$), D ($E - F$), each time restoring a degree-2 alphabetically-first candidate while F, G stay at degree 3. Then E has neighbours F, G with $F - G$ already present, leaving the triangle F, G, H, from which F, G, H are eliminated at degrees $2, 1, 0$. (Check!)
The degrees encountered are $2, 2, 2, 2, 2, 2, 1, 0$.

Problem 6.10. What is the treewidth of the network in Figure 6.11? Justify your answer.
Figure 6.11 is the Bayesian network structure on the variables A, B, C, D, E, F, G, H with edges

$$A \to B, \quad A \to C, \quad B \to D, \quad C \to E,$$
$$D \to F, \quad E \to G, \quad F \to H, \quad G \to H,$$

so the two chains $A \to B \to D \to F$ and $A \to C \to E \to G$ leave the root A and rejoin at the sink H, whose parents are F and G.

Solution. The treewidth is 2.
Upper bound. Exercise 6.8 exhibits the complete order A, B, C, D, E, F, G, H of width 2, and treewidth is the minimum width over complete orders.
Lower bound. Widths are measured on the moral graph, in which every family is a clique; here H has parents F, G, so $\{F, G, H\}$ is a triangle ($F - H$, $G - H$ from the DAG, $F - G$ from moralization). Elimination never breaks a clique, since its only structural effects are adding fill-in edges and deleting the eliminated variable. So for any complete order π, whichever of F, G, H comes first is still adjacent

to the other two when eliminated, giving it degree ≥ 2 and π width ≥ 2. (This is the case $k = 2$ of Exercise 6.15.)

Problem 6.11. Given a Bayesian network that has a tree structure with at least one edge, describe an efficient algorithm for obtaining an elimination order that is guaranteed to have width 1.
(Recall the terminology of Section 6.8: a **tree network** is a polytree in which every node has at most one parent, and such networks are stated there to have treewidth at most 1.)

Solution. Reverse any topological order $X_1, \ldots, X_n$ of the network: $\pi = X_n, X_{n-1}, \ldots, X_1$, i.e. peel leaves inward toward the roots. Depth-first search produces a topological order in $O(n+m)$ time, and $m \leq n$ in a tree network, so this is linear in the network structure.
Its width is 1. No node has two parents, so moralization adds nothing and the interaction graph G_0 (Definition 6.4) is the undirected version of the DAG, a forest. When OrderWidth (Algorithm 4) reaches X_i, the current graph is G_0 restricted to $\{X_1, \ldots, X_i\}$ with no fill-in ever added: the neighbours of X_i in G_0 are its unique parent and its children, and every child follows X_i topologically, hence precedes it in π and is already gone. So X_i has at most its parent as a neighbour, connecting a set of size ≤ 1 adds no edge, and deleting X_i leaves G_0 on $\{X_1, \ldots, X_{i-1}\}$ – the same statement one step down. Hence $d \leq 1$ at every step, with $d = 1$ at any node having a parent, one of which exists since the network has at least one edge.

Problem 6.12. Given an elimination order π that covers all network variables, show how we can use this particular order and Algorithm 8, `VE_PR`, to compute the marginal $Pr(X, \mathbf{e})$ for every network variable X and some evidence $\mathbf{e}$. What is the total complexity of computing all such marginals?
(Algorithm 8, `VE_PR`, takes a network $\mathcal{N}$, query variables $\mathbf{Q}$, and evidence $\mathbf{e}$. It first prunes the network given the query, then chooses an order σ of the variables not in $\mathbf{Q}$, reduces every CPT of the pruned network on $\mathbf{e}$, eliminates the variables of σ in turn by multiplying all factors that mention the current variable and summing it out, and finally returns the product of the surviving factors, which equals $Pr(\mathbf{Q}, \mathbf{e})$.)

Solution. Call VE_PR($\mathcal{N}, \{X\}, \mathbf{e}$) once per variable $X = v_k$, supplying on Line 2 the order

$$\pi_X \;=\; v_1, \ldots, v_{k-1}, v_{k+1}, \ldots, v_n$$

restricted to the variables surviving the pruning on Line 1; Line 2 admits any order of the variables outside $\mathbf{Q}$, so each call is correct and returns $Pr(X, \mathbf{e})$. Total cost $O(n^2 \exp(w))$ time and $O(n \exp(w))$ space, where w is the width of π.
The cost follows from two facts about widths, both measured on interaction graphs (Definition 6.4), and both proved by running the elimination twice in parallel on graphs G_i (for the full order) and H_i (for the restricted one).
(i) *Deleting one variable from an order costs at most* 1. The runs of π and π_X on the same graph satisfy the invariant that every edge of H_i avoiding X is an edge of G_i: the step $v_i = X$ only adds edges on the G side, and at a step $v_i \neq X$ a fill-in edge $\{Y, Z\}$ of H_{i+1} has Y, Z adjacent to v_i in H_i, hence in G_i by hypothesis, so the same edge appears in G_{i+1}. Applying the invariant to the pairs $\{v_i, Y\}$, the neighbours of v_i in H_i are among those in G_i plus possibly X, so $|N_{H_i}(v_i)| \leq |N_{G_i}(v_i)| + 1 \leq w + 1$.
(ii) *Restricting to a subgraph never hurts.* For G' a subgraph of G on $S \subseteq V(G)$ and σ the restriction of τ to S, the same induction with the invariant $E(H_i) \subseteq E(G_i)$ shows σ has width at most that of τ: steps eliminating a variable outside S only add edges on the G side, and at a step $v_i \in S$ the H-neighbours of v_i are G-neighbours, so the recorded degree is no larger and every H-fill-in is a G-fill-in.
Since pruneNetwork only deletes nodes and edges and shrinks CPTs, the moral graph of the pruned network is such a subgraph, so by (i) and (ii) each call runs on an order of width at most $w + 1$. With $|\mathbf{Q}| = 1$, Theorem 6.2 bounds each call by $O(n \exp(w + 1))$, pruning costing only $O(n \exp(w))$ more; over n variables this is $O(n^2 \exp(w + 1)) = O(n^2 \exp(w))$ for bounded variable cardinalities. The n calls are independent, so they reuse the same $O(n \exp(w))$ space, retaining only the n single-variable answers.

Problem 6.13. Prove the following: for factors f_i, f_j, and evidence $\mathbf{e}$,

$$(f_i f_j)^{\mathbf{e}} \;=\; f_i^{\mathbf{e}}\, f_j^{\mathbf{e}}.$$

(This is Theorem 6.3. Recall Definition 6.3: the product of factors $f_1(\mathbf{X})$ and $f_2(\mathbf{Y})$ is the factor over $\mathbf{Z} = \mathbf{X} \cup \mathbf{Y}$ given by $(f_1 f_2)(\mathbf{z}) = f_1(\mathbf{x}) f_2(\mathbf{y})$, where $\mathbf{x}$ and $\mathbf{y}$ are the instantiations compatible with $\mathbf{z}$; and Definition 6.5: the reduction of a factor $f(\mathbf{X})$ given evidence $\mathbf{e}$ is the factor $f^{\mathbf{e}}$ over $\mathbf{X}$ with $f^{\mathbf{e}}(\mathbf{x}) = f(\mathbf{x})$ if $\mathbf{x} \sim \mathbf{e}$ and $f^{\mathbf{e}}(\mathbf{x}) = 0$ otherwise.)

Solution. Both sides are factors over $\mathbf{Z} = \mathbf{X} \cup \mathbf{Y}$ (Definition 6.3 for the product, and reduction preserves a factor's variables), so it suffices to compare their values at each instantiation $\mathbf{z}$. Fix $\mathbf{z}$ and let $\mathbf{x}, \mathbf{y}$ be its restrictions to $\mathbf{X}, \mathbf{Y}$, so that Definition 6.3 gives $(f_i f_j)(\mathbf{z}) = f_i(\mathbf{x}) f_j(\mathbf{y})$ and $(f_i^{\mathbf{e}} f_j^{\mathbf{e}})(\mathbf{z}) = f_i^{\mathbf{e}}(\mathbf{x}) f_j^{\mathbf{e}}(\mathbf{y})$.
(i) $\mathbf{z} \sim \mathbf{e}$. Any variable shared by $\mathbf{X}$ and $\mathbf{e}$ is shared by $\mathbf{Z}$ and $\mathbf{e}$, where $\mathbf{x}$ copies $\mathbf{z}$, so $\mathbf{x} \sim \mathbf{e}$, and likewise $\mathbf{y} \sim \mathbf{e}$. Definition 6.5 then leaves all three values unreduced:

$$(f_i f_j)^{\mathbf{e}}(\mathbf{z}) = f_i(\mathbf{x}) f_j(\mathbf{y}) = f_i^{\mathbf{e}}(\mathbf{x})\, f_j^{\mathbf{e}}(\mathbf{y}).$$

(ii) $\mathbf{z} \not\sim \mathbf{e}$. The left side is 0. Some variable V of $\mathbf{Z}$ is instantiated differently by $\mathbf{z}$ and $\mathbf{e}$, and V lies in $\mathbf{X}$ or $\mathbf{Y}$; in the first case $\mathbf{x}$ inherits that value, so $\mathbf{x} \not\sim \mathbf{e}$ and $f_i^{\mathbf{e}}(\mathbf{x}) = 0$, in the second $f_j^{\mathbf{e}}(\mathbf{y}) = 0$. Either way the right side vanishes too.

Problem 6.14. Prove the following: if $\mathcal{N}' = \text{pruneEdges}(\mathcal{N}, \mathbf{e})$, then $Pr(\mathbf{Q}, \mathbf{e}) = Pr'(\mathbf{Q}, \mathbf{e})$ for every $\mathbf{Q}$, where Pr and Pr' are the distributions induced by $\mathcal{N}$ and $\mathcal{N}'$. Hint: prove that the joint distributions of $\mathcal{N}$ and $\mathcal{N}'$ agree on rows that are consistent with evidence $\mathbf{e}$.
(This is Theorem 6.5. Recall the edge-pruning operation of Section 6.9.2: for each edge $U \to X$ whose tail U is a variable of the evidence $\mathbf{e}$, we delete the edge from the network and replace the CPT $\Theta_{X|\mathbf{U}}$ of X by the smaller table obtained from it by fixing U to the value u that $\mathbf{e}$ assigns it, namely $\sum_U \Theta^u_{X|\mathbf{U}}$.)

Solution. Following the hint, Pr and Pr' agree on every row consistent with $\mathbf{e}$, and the two marginals sum over the same such rows.
Write $\mathbf{U}_X$ for the parents of X in $\mathcal{N}$, split as $\mathbf{U}_X^{\mathbf{E}} = \mathbf{U}_X \cap \mathbf{E}$ and $\mathbf{U}_X^{-} = \mathbf{U}_X \setminus \mathbf{E}$, and let $\mathbf{e}_X$ be the restriction of $\mathbf{e}$ to $\mathbf{U}_X^{\mathbf{E}}$. Pruning deletes exactly the edges out of evidence variables, so $\mathcal{N}'$ has the same nodes, X has parents $\mathbf{U}_X^{-}$, and deleting edges preserves acyclicity. Its new CPT is $\theta'_{x|\mathbf{u}^-} = \theta_{x|\mathbf{u}^-\mathbf{e}_X}$, a legitimate CPT since $\mathbf{u}^-\mathbf{e}_X$ is a full instantiation of $\mathbf{U}_X$ and so $\sum_x \theta_{x|\mathbf{u}^-\mathbf{e}_X} = 1$.
Let $\mathbf{z}$ be an instantiation of all variables with $\mathbf{z} \sim \mathbf{e}$, i.e. extending $\mathbf{e}$, and let $x, \mathbf{u}_X, \mathbf{u}_X^-$ be the values it assigns. It gives the evidence parents exactly $\mathbf{e}_X$, so $\mathbf{u}_X = \mathbf{u}_X^- \mathbf{e}_X$ and, by the chain rule in each network,

$$Pr'(\mathbf{z}) = \prod_{X \in \mathbf{Z}} \theta'_{x|\mathbf{u}_X^-} = \prod_{X \in \mathbf{Z}} \theta_{x|\mathbf{u}_X} = Pr(\mathbf{z}).$$

Now fix $\mathbf{q}$. If $\mathbf{q} \not\sim \mathbf{e}$ both marginals are 0. Otherwise both are sums over the same instantiations $\mathbf{z}$ extending $\mathbf{qe}$,

$$Pr(\mathbf{q}, \mathbf{e}) = \sum_{\mathbf{z} \sim \mathbf{qe}} Pr(\mathbf{z}) = \sum_{\mathbf{z} \sim \mathbf{qe}} Pr'(\mathbf{z}) = Pr'(\mathbf{q}, \mathbf{e}),$$

each such $\mathbf{z}$ extending $\mathbf{e}$ and so covered by the previous display.

5.3 Exercises 6.15–6.18

Problem 6.15. Suppose we have a Bayesian network containing a node with k parents. Show that every variable order that contains all network variables must have width at least k.

Solution. Let X have parents $U_1, \ldots, U_k$, put $C = \{X, U_1, \ldots, U_k\}$, and let π be any order over all network variables.
C is a clique of the initial interaction graph G_1: the CPT $\Theta_{X|U_1 \ldots U_k}$ is a factor over exactly C, and by Definition 6.4 variables sharing a factor are pairwise adjacent. Elimination never destroys an edge between two surviving variables, since it only adds fill-in edges and removes edges incident on the eliminated variable (Section 6.6).
Let i be least with $\pi(i) \in C$; it exists since π covers every variable. Steps $1, \ldots, i-1$ all eliminate variables outside C, so the other k members of C survive into G_i and remain adjacent to $\pi(i)$. Hence $\pi(i)$ has at least k neighbours when eliminated, so the d recorded by OrderWidth at iteration i is at least k and

$$w \;=\; \max_j d_j \;\geq\; k.$$

Problem 6.16. Consider the naive Bayes structure $C \to A_1, \ldots, C \to A_m$, where all variables are binary. That is, the network has a single root C (the class variable) with CPT Θ_C, and m leaves $A_1, \ldots, A_m$ (the attributes), where each A_i has C as its only parent and CPT $\Theta_{A_i|C}$. Show how the algorithm of variable elimination can be used to compute a closed form for the conditional probability $\Pr(c \mid a_1, \ldots, a_m)$.

Solution.

$$\Pr(c \mid a_1, \ldots, a_m) = \frac{\theta_c \prod_{i=1}^m \theta_{a_i|c}}{\theta_c \prod_{i=1}^m \theta_{a_i|c} \;+\; \theta_{\bar{c}} \prod_{i=1}^m \theta_{a_i|\bar{c}}},$$

where $c, \bar{c}$ are the values of C. Run $\mathrm{VE_{PR}}$ with $Q = \{C\}$, $e : A_1 = a_1, \ldots, A_m = a_m$ and order $\pi = A_1, \ldots, A_m$. Line 3 reduces each CPT (Definition 6.5), leaving Θ_C untouched and giving

$$\Theta^e_{A_i|C}(\alpha, \gamma) = \begin{cases} \theta_{a_i|\gamma}, & \alpha = a_i, \\ 0, & \alpha \neq a_i, \end{cases} \qquad \gamma \in \{c, \bar{c}\},$$

which is legitimate because reduction distributes over the product of CPTs (Theorem 6.3). Each A_i occurs in the single factor $\Theta^e_{A_i|C}$, so its elimination needs no multiplication and yields $f_i(\gamma) = \theta_{a_i|\gamma}$, a factor over C alone (width 1, so the run costs $O(m)$). Line 9 multiplies the survivors $\Theta_C, f_1, \ldots, f_m$:

$$\Pr(C, e)(c) \;=\; \theta_c \prod_{i=1}^m \theta_{a_i|c}, \qquad \Pr(C, e)(\bar{c}) \;=\; \theta_{\bar{c}} \prod_{i=1}^m \theta_{a_i|\bar{c}},$$

and dividing by $\Pr(e) = \Pr(C, e)(c) + \Pr(C, e)(\bar{c})$ gives the displayed form.

Problem 6.17. Suppose we have a Bayesian network $\mathcal{N}$ and evidence e with some node X having parents $U_1, \ldots, U_n$ (all binary).

- (a) Show that if X is a logical OR node that is set to false in e, we can prune all edges incoming into X and assert hard evidence on each parent.
- (b) Show that if X is a noisy OR node that is set to false in e, we can prune all edges incoming into X and assert soft evidence on each parent.

In particular, for each case identify a network $\mathcal{N}'$ and evidence e' where all edges incoming into X have been pruned from $\mathcal{N}$ and where $\Pr(\mathbf{x}, e) = \Pr'(\mathbf{x}, e')$ for every network instantiation $\mathbf{x}$.
Here a **logical OR** node is one whose CPT satisfies $\theta_{x|\mathbf{u}} = 1$ iff at least one U_i is true in $\mathbf{u}$, and a

noisy OR node is one whose CPT is generated by the noisy-or model of Section 5.4.1, so that with suppressor probabilities $\theta_{q_1}, \ldots, \theta_{q_n}$ and leak probability θ_l,

$$\theta_{\bar{x}|\mathbf{u}} \;=\; (1-\theta_l) \prod_{i\,:\,U_i=\text{true in } \mathbf{u}} \theta_{q_i}.$$

Solution. Both parts turn on one fact: with X set to false, the row $\theta_{\bar{x}|\mathbf{u}}$ of an OR-like CPT factorizes into unary functions of the parents, and a unary factor on U_i is exactly what evidence on U_i contributes. Write $\mathbf{z}$ for an instantiation of $\mathcal{N}$'s variables (the exercise's $\mathbf{x}$), $\mathbf{u}$ for its restriction to the parents, $\bar{x}$ for $X = \text{false}$, and $R(\mathbf{z}) = \prod_{V \neq X} \theta_{z_V|\mathbf{z}_{\mathrm{Pa}(V)}}$ for the chain-rule product over the other CPTs, which nothing below touches.

(a) Logical OR. Take $\mathcal{N}'$ with the edges $U_i \to X$ deleted and $\Theta_{X|U_1\ldots U_n}$ replaced by the root CPT $\theta'_{\bar{x}} = 1$, $\theta'_{x} = 0$, and take $e' = e \cup \{U_1 = \text{false}, \ldots, U_n = \text{false}\}$; deleting edges preserves acyclicity. Since X is false iff every parent is,

$$\theta_{\bar{x}|\mathbf{u}} = \prod_{i=1}^{n} \big[\, U_i = \text{false in } \mathbf{u} \,\big],$$

with $[\cdot]$ the 0/1 indicator. If $\mathbf{z}$ is inconsistent with e both sides vanish, e' extending e; otherwise $z_X = \bar{x}$ and

$$\Pr(\mathbf{z}, e) = \theta_{\bar{x}|\mathbf{u}}\, R(\mathbf{z}) = R(\mathbf{z}) \prod_{i=1}^{n} \big[\, z_{U_i} = \text{false} \,\big] = \acute{\Pr}(\mathbf{z}, e'),$$

since $\Pr'(\mathbf{z}) = R(\mathbf{z})$ and $\mathbf{z} \sim e'$ iff every z_{U_i} is false. (If e already sets some U_i true then $\Pr(e) = 0$ and the claim is trivial.)

(b) Noisy OR. By (5.2) of Section 5.4.1,

$$\theta_{\bar{x}|\mathbf{u}} = (1-\theta_l) \prod_{i=1}^{n} g_i(z_{U_i}), \qquad g_i(\text{true}) = \theta_{q_i}, \quad g_i(\text{false}) = 1,$$

and the g_i are no longer 0/1, so they are likelihoods rather than hard evidence; realize them as virtual evidence (Section 3.6.4). Let $\mathcal{N}'$ delete the edges $U_i \to X$, give the now-root X the CPT $\theta'_{\bar{x}} = 1 - \theta_l$, $\theta'_{x} = \theta_l$, and attach to each U_i a fresh binary child V_i with

$$\theta'_{v_i|u_i} = \theta_{q_i}, \qquad \theta'_{v_i|\bar{u}_i} = 1,$$

the complementary rows being $1 - \theta_{q_i}$ and 0; set $e' = e \cup \{V_1 = v_1, \ldots, V_n = v_n\}$. Both new tables are CPTs (rows nonnegative and summing to 1). Since e' fixes every V_i, each $\mathbf{z}$ extends to exactly one instantiation of $\mathcal{N}'$ consistent with e'. For $\mathbf{z}$ consistent with e, so that $z_X = \bar{x}$, the chain rule over the enlarged variable set gives

$$\acute{\Pr}(\mathbf{z}, e') = \theta'_{\bar{x}} \cdot R(\mathbf{z}) \cdot \prod_{i=1}^{n} \theta'_{v_i|z_{U_i}} = (1-\theta_l)\, R(\mathbf{z}) \prod_{i=1}^{n} g_i(z_{U_i}) = \Pr(\mathbf{z}, e),$$

because $\theta'_{v_i|z_{U_i}}$ is θ_{q_i} when z_{U_i} is true and 1 when it is false, i.e. $g_i(z_{U_i})$; and both sides are 0 when $\mathbf{z} \not\sim e$.

Problem 6.18. Suppose that f_1 is a factor over variables XY and f_2 is a factor over variables XZ. Show that $(\sum_X f_1)\,(\sum_X f_2)$ is an upper bound on $\sum_X f_1 f_2$ in the following sense:

$$\left(\left(\sum_X f_1\right)\left(\sum_X f_2\right)\right)(\mathbf{u}) \;\geq\; \left(\sum_X f_1 f_2\right)(\mathbf{u}),$$

for all instantiations $\mathbf{u}$ of $\mathbf{U} = \mathbf{Y} \cup \mathbf{Z}$. Show how this property can be used to produce an approximate

version of VE_PR that can have a lower complexity yet is guaranteed to produce an upper bound on the true marginals.

Solution. The gap is the off-diagonal mass, which is nonnegative because factors are nonnegative (Definition 6.1). Fix $\mathbf{u}$ with restrictions $\mathbf{y}, \mathbf{z}$ to $\mathbf{Y}, \mathbf{Z}$. By Definitions 6.2 and 6.3 the left side decouples the two sums,

$$\left(\left(\sum_X f_1\right)\left(\sum_X f_2\right)\right)(\mathbf{u}) = \sum_x \sum_{x'} f_1(x, \mathbf{y})\, f_2(x', \mathbf{z}),$$

whereas $\left(\sum_X f_1 f_2\right)(\mathbf{u}) = \sum_x f_1(x, \mathbf{y}) f_2(x, \mathbf{z})$ is its diagonal, so the difference is $\sum_{x \neq x'} f_1(x, \mathbf{y}) f_2(x', \mathbf{z}) \geq 0$.

Mini-buckets. For nonnegative factors, $f \geq \hat{f}$ pointwise implies $fg \geq \hat{f}g$ and $\sum_W f \geq \sum_W \hat{f}$, so replacing a factor set by one whose product dominates pointwise makes the final result of VE_PR dominate, by induction on the remaining eliminations. Fix a bound b on the variables allowed in a constructed factor. At iteration i, partition the factors mentioning $\pi(i)$ into groups $P_1, \ldots, P_r$ with

$$\left| \bigcup_{f \in P_j} \mathrm{vars}(f) \right| \leq b + 1,$$

and replace them all by the r factors $\hat{f}_{i,j} = \sum_{\pi(i)} \prod_{f \in P_j} f$ rather than by the exact f_i. Applying the inequality $r - 1$ times, each application decoupling one more copy of the summation variable,

$$\prod_{j=1}^{r} \hat{f}_{i,j} \geq \sum_{\pi(i)} \prod_{j=1}^{r} \prod_{f \in P_j} f = f_i,$$

so by monotonicity Line 9 returns $\widehat{\Pr}(\mathbf{q}, e) \geq \Pr(\mathbf{q}, e)$ for every $\mathbf{q}$, and $\mathbf{Q} = \emptyset$ bounds $\Pr(e)$. Every factor built has at most b variables, so the cost is $O(n \exp(b))$ rather than $O(n \exp(w))$, with $b \geq w$ recovering exact VE_PR. The guarantee is on the joint $\Pr(\mathbf{Q}, e)$, not on the conditional: normalizing divides two upper bounds.

A worked instance. Take X, Y, Z binary with

x	y	$f_1(x, y)$
0	0	0.2
0	1	0.5
1	0	0.4
1	1	0.3

x	z	$f_2(x, z)$
0	0	0.3
0	1	0.6
1	0	0.6
1	1	0.2

Summing out X gives $\left(\sum_X f_1\right) = (0.6, 0.8)$ and $\left(\sum_X f_2\right) = (0.9, 0.8)$, so

y	z	$\left(\sum_X f_1\right)\left(\sum_X f_2\right)$	$\sum_X f_1 f_2$	gap
0	0	0.54	0.30	0.24
0	1	0.48	0.20	0.28
1	0	0.72	0.33	0.39
1	1	0.64	0.36	0.28

and each gap is the off-diagonal sum, e.g. $0.2 \cdot 0.6 + 0.4 \cdot 0.3 = 0.24$ at $y = z = 0$.

6 Inference by Factor Elimination

Contents

6.1 Exercises 7.1–7.7 82
6.2 Exercises 7.8–7.14 90
6.3 Exercises 7.15–7.15 95

6.1 Exercises 7.1–7.7

Problem 7.1. Answer the following queries with respect to the Bayesian network in Figure 7.18:

1. $\Pr(B, C)$.
2. $\Pr(C, D = \text{true})$.
3. $\Pr(A \mid D = \text{true}, E = \text{true})$.

You may prune the network before attempting each computation and use any inference method you find most appropriate.

The network of Figure 7.18 has seven binary variables A, B, C, D, E, F, G and edges

$$A \to C, \quad B \to C, \quad C \to D, \quad C \to E, \quad D \to F, \quad E \to G.$$

Its CPTs are the following.

A	f_A
true	.6
false	.4

B	f_B
true	.5
false	.5

A	B	C	f_C
true	true	true	.9
true	true	false	.1
true	false	true	.1
true	false	false	.9
false	true	true	.5
false	true	false	.5
false	false	true	.3
false	false	false	.7

C	D	f_D
true	true	.2
true	false	.8
false	true	.7
false	false	.3

C	E	f_E
true	true	.1
true	false	.9
false	true	.2
false	false	.8

D	F	f_F
true	true	.7
true	false	.3
false	true	.6
false	false	.4

E	G	f_G
true	true	.2
true	false	.8
false	true	.8
false	false	.2

Solution. Write $f_A = \Theta_A$, $f_B = \Theta_B$, $f_C = \Theta_{C|AB}$, $f_D = \Theta_{D|C}$, $f_E = \Theta_{E|C}$, $f_F = \Theta_{F|D}$, $f_G = \Theta_{G|E}$.

Summing $f_A f_B f_C$ over B once gives the joint over A, C that all three parts use:

$$\begin{aligned}\Pr(A=t, C=t) &= .6\,(.5\cdot.9 + .5\cdot.1) = .30,\\ \Pr(A=t, C=f) &= .6\,(.5\cdot.1 + .5\cdot.9) = .30,\\ \Pr(A=f, C=t) &= .4\,(.5\cdot.5 + .5\cdot.3) = .16,\\ \Pr(A=f, C=f) &= .4\,(.5\cdot.5 + .5\cdot.7) = .24,\end{aligned}$$

so $\Pr(C=t) = .46$ and $\Pr(C=f) = .54$.
Part 1. With $\mathbf{Q} = \{B, C\}$ and no evidence, node pruning deletes the leaves F, G and then D, E, leaving $A \to C \leftarrow B$, so $\Pr(B, C) = \sum_A f_A f_B f_C$:

$$\begin{aligned}\Pr(B=t, C=t) &= .5\,(.6\cdot.9 + .4\cdot.5) = .37,\\ \Pr(B=t, C=f) &= .5\,(.6\cdot.1 + .4\cdot.5) = .13,\\ \Pr(B=f, C=t) &= .5\,(.6\cdot.1 + .4\cdot.3) = .09,\\ \Pr(B=f, C=f) &= .5\,(.6\cdot.9 + .4\cdot.7) = .41.\end{aligned}$$

B	C	$\Pr(B, C)$
true	true	.37
true	false	.13
false	true	.09
false	false	.41

Part 2. With $\mathbf{Q} = \{C\}$, $\mathbf{e} : D = \text{true}$, node pruning deletes F, G and then E, and edge pruning removes nothing (D is childless), leaving $A \to C \leftarrow B$ with $C \to D$. Hence $\Pr(C, D=t) = \Pr(C)\,\Theta_{D=t|C}$:

$$\Pr(C=t, D=t) = .46\cdot.2 = .092, \qquad \Pr(C=f, D=t) = .54\cdot.7 = .378.$$

C	$\Pr(C, D = \text{true})$
true	.092
false	.378

Part 3. With $\mathbf{Q} = \{A\}$, $\mathbf{e} : D = E = \text{true}$, pruning deletes only F, G. Put $g(C) = \Theta_{D=t|C}\,\Theta_{E=t|C}$, so $g(t) = .02$ and $g(f) = .14$; summing out C against the joint above,

$$\begin{aligned}\Pr(A=t, D=t, E=t) &= .30\cdot.02 + .30\cdot.14 = .048,\\ \Pr(A=f, D=t, E=t) &= .16\cdot.02 + .24\cdot.14 = .0368,\end{aligned}$$

so $\Pr(D=t, E=t) = .0848$ and dividing gives $.048/.0848 = 30/53$, $.0368/.0848 = 23/53$.

A	$\Pr(A \mid D = \text{true}, E = \text{true})$
true	$30/53 \approx .5660$
false	$23/53 \approx .4340$

Problem 7.2. Consider the Bayesian network in Figure 7.19. Construct an elimination tree for the Bayesian network CPTs that has the smallest width possible and assigns at most one CPT to each tree node. Compute the separators, clusters, and width of the elimination tree. It may be useful to know that this network has the following jointree:

$$ABC - BCE - BDE - DEF - EFG - FGH.$$

The DAG of Figure 7.19 has eight binary variables and the edges

$$\begin{aligned}&A \to B, \quad A \to C, \quad B \to D, \quad B \to E, \quad C \to E,\\ &D \to F, \quad E \to F, \quad E \to G, \quad F \to H, \quad G \to H.\end{aligned}$$

Its CPTs are therefore Θ_A, $\Theta_{B|A}$, $\Theta_{C|A}$, $\Theta_{D|B}$, $\Theta_{E|BC}$, $\Theta_{F|DE}$, $\Theta_{G|E}$, and $\Theta_{H|FG}$.

Solution. Take T to be the path

$$3 - 2 - 1 - 4 - 5 - 6 - 7 - 8$$

with the one-to-one assignment of CPTs to nodes tabulated below; its width is 2, the smallest attainable, since $\Theta_{E|BC}$ must sit alone at some node i, forcing $\{B, C, E\} = \text{vars}(i) \subseteq \mathbf{C}_i$.

node	factor ϕ_i	vars(i)
3	Θ_A	A
2	$\Theta_{B\|A}$	AB
1	$\Theta_{C\|A}$	AC
4	$\Theta_{E\|BC}$	BCE
5	$\Theta_{D\|B}$	BD
6	$\Theta_{F\|DE}$	DEF
7	$\Theta_{G\|E}$	EG
8	$\Theta_{H\|FG}$	FGH

This is an elimination tree by Definition 7.2 (T is a tree and each of the eight CPTs is assigned to exactly one node). Since T is a path, each edge splits it into an initial and a final segment, so Definition 7.3 gives

$$\begin{aligned}
\mathbf{S}_{32} &= \{A\} \cap \{A, B, C, D, E, F, G, H\} = \{A\}, \\
\mathbf{S}_{21} &= \{A, B\} \cap \{A, B, C, D, E, F, G, H\} = \{A, B\}, \\
\mathbf{S}_{14} &= \{A, B, C\} \cap \{B, C, D, E, F, G, H\} = \{B, C\}, \\
\mathbf{S}_{45} &= \{A, B, C, E\} \cap \{B, D, E, F, G, H\} = \{B, E\}, \\
\mathbf{S}_{56} &= \{A, B, C, D, E\} \cap \{D, E, F, G, H\} = \{D, E\}, \\
\mathbf{S}_{67} &= \{A, B, C, D, E, F\} \cap \{E, F, G, H\} = \{E, F\}, \\
\mathbf{S}_{78} &= \{A, B, C, D, E, F, G\} \cap \{F, G, H\} = \{F, G\}.
\end{aligned}$$

and Definition 7.4, $\mathbf{C}_i = \text{vars}(i) \cup \bigcup_j \mathbf{S}_{ij}$, gives

$$\begin{aligned}
\mathbf{C}_3 &= \{A\} \cup \{A\} = \{A\}, \\
\mathbf{C}_2 &= \{A, B\} \cup \{A\} \cup \{A, B\} = \{A, B\}, \\
\mathbf{C}_1 &= \{A, C\} \cup \{A, B\} \cup \{B, C\} = \{A, B, C\}, \\
\mathbf{C}_4 &= \{B, C, E\} \cup \{B, C\} \cup \{B, E\} = \{B, C, E\}, \\
\mathbf{C}_5 &= \{B, D\} \cup \{B, E\} \cup \{D, E\} = \{B, D, E\}, \\
\mathbf{C}_6 &= \{D, E, F\} \cup \{D, E\} \cup \{E, F\} = \{D, E, F\}, \\
\mathbf{C}_7 &= \{E, G\} \cup \{E, F\} \cup \{F, G\} = \{E, F, G\}, \\
\mathbf{C}_8 &= \{F, G, H\} \cup \{F, G\} = \{F, G, H\}.
\end{aligned}$$

The largest cluster has three variables, so the width is $3 - 1 = 2$, matching the lower bound above.

Problem 7.3. Consider the Bayesian network in Figure 7.18 and the corresponding elimination tree in Figure 7.20, and suppose that the evidence indicator for each variable is assigned to the node corresponding to that variable (e.g., λ_C is assigned to node 3). Suppose we are answering the following queries according to the given order:

$$\Pr(G = \text{true}), \quad \Pr(G, F = \text{true}), \quad \Pr(F, A = \text{true}, F = \text{true}),$$

using Algorithm 12, FE.

1. Compute the separators, clusters, and width of the given elimination tree.
2. What messages are computed while answering each query according to the previous sequence? State the origin, destination, and value of each message. Use node 7 as the root to answer the first two queries, and node 6 as the root to answer the last query. For each query, compute only the messages directed toward the corresponding root.

3. What messages are invalidated due to new evidence as we attempt each new query?
4. What is the answer to each of the previous queries?

The Bayesian network of Figure 7.18 is the one described in Exercise 7.1: variables A, B, C, D, E, F, G, edges $A \to C$, $B \to C$, $C \to D$, $C \to E$, $D \to F$, $E \to G$, and CPTs $f_A = \Theta_A = (.6, .4)$, $f_B = \Theta_B = (.5, .5)$, $f_C = \Theta_{C|AB}$ with $\Theta_{C=t|A=t,B=t} = .9$, $\Theta_{C=t|A=t,B=f} = .1$, $\Theta_{C=t|A=f,B=t} = .5$, $\Theta_{C=t|A=f,B=f} = .3$, and

$$\begin{aligned}
\Theta_{D=t|C=t} &= .2, \quad \Theta_{D=t|C=f} = .7, \\
\Theta_{E=t|C=t} &= .1, \quad \Theta_{E=t|C=f} = .2, \\
\Theta_{F=t|D=t} &= .7, \quad \Theta_{F=t|D=f} = .6, \\
\Theta_{G=t|E=t} &= .2, \quad \Theta_{G=t|E=f} = .8.
\end{aligned}$$

The elimination tree of Figure 7.20 has seven nodes, one per CPT, namely $\phi_1 = f_A$, $\phi_2 = f_B$, $\phi_3 = f_C$, $\phi_4 = f_D$, $\phi_5 = f_E$, $\phi_6 = f_F$, $\phi_7 = f_G$, and the edges

$$1-3, \quad 2-3, \quad 3-4, \quad 3-5, \quad 4-6, \quad 5-7.$$

(So node 3 has neighbours $1, 2, 4, 5$; node 4 has neighbours $3, 6$; node 5 has neighbours $3, 7$; nodes $1, 2, 6, 7$ are leaves.) The evidence indicators are assigned as λ_A to node 1, λ_B to node 2, λ_C to node 3, λ_D to node 4, λ_E to node 5, λ_F to node 6, λ_G to node 7.

Solution. Part 1. Width 2. With $\text{vars}(1) = \{A\}$, $\text{vars}(2) = \{B\}$, $\text{vars}(3) = \{A, B, C\}$, $\text{vars}(4) = \{C, D\}$, $\text{vars}(5) = \{C, E\}$, $\text{vars}(6) = \{D, F\}$, $\text{vars}(7) = \{E, G\}$ (each indicator λ_X sits at a node already containing X, so it adds no variable), Definition 7.3 gives

$$\begin{aligned}
\mathbf{S}_{13} &= \{A\} \cap \{A, B, C, D, E, F, G\} = \{A\}, \\
\mathbf{S}_{23} &= \{B\} \cap \{A, B, C, D, E, F, G\} = \{B\}, \\
\mathbf{S}_{34} &= \{A, B, C, E, G\} \cap \{C, D, F\} = \{C\}, \\
\mathbf{S}_{35} &= \{A, B, C, D, F\} \cap \{C, E, G\} = \{C\}, \\
\mathbf{S}_{46} &= \{A, B, C, D, E, G\} \cap \{D, F\} = \{D\}, \\
\mathbf{S}_{57} &= \{A, B, C, D, E, F\} \cap \{E, G\} = \{E\}.
\end{aligned}$$

and Definition 7.4, $\mathbf{C}_i = \text{vars}(i) \cup \bigcup_j \mathbf{S}_{ij}$, gives

$$\begin{aligned}
&\mathbf{C}_1 = \{A\}, \quad && \mathbf{C}_2 = \{B\}, \quad && \mathbf{C}_3 = \{A, B, C\}, \\
&\mathbf{C}_4 = \{C, D\}, && \mathbf{C}_5 = \{C, E\}, && \mathbf{C}_6 = \{D, F\}, \\
&\mathbf{C}_7 = \{E, G\}. &&
\end{aligned}$$

The largest cluster is $\mathbf{C}_3$, of size three, so the width is $3 - 1 = 2$.
Part 2. Messages come from Equation 7.2, $M_{ij} = \text{project}(\phi_i \prod_{k \neq j} M_{ki}, \mathbf{S}_{ij})$, with ϕ_i including node i's indicators (a λ_X for X outside the current evidence is the constant 1).
Query 1: $\Pr(G = \text{true})$, evidence $G = \text{true}$, root 7. The messages toward 7 are

$$\begin{aligned}
M_{13}(A) &= f_A \lambda_A = (.6, .4), \\
M_{23}(B) &= f_B \lambda_B = (.5, .5), \\
M_{64}(D) &= \textstyle\sum_F f_F \lambda_F = (1, 1), \\
M_{43}(C) &= \textstyle\sum_D f_D \lambda_D M_{64} = (1, 1), \\
M_{35}(C) &= \textstyle\sum_{A,B} f_C \lambda_C M_{13} M_{23} M_{43}, \\
M_{57}(E) &= \textstyle\sum_C f_E \lambda_E M_{35},
\end{aligned}$$

over A, B, D, C, C, E respectively, matching the separators of Part 1. Numerically $M_{35} = \Pr(C)$:

$$\begin{aligned} M_{35}(C=t) &= .6(.5\cdot.9 + .5\cdot.1) + .4(.5\cdot.5 + .5\cdot.3) \\ &= .30 + .16 = .46, \\ M_{35}(C=f) &= .6(.5\cdot.1 + .5\cdot.9) + .4(.5\cdot.5 + .5\cdot.7) \\ &= .30 + .24 = .54, \end{aligned}$$

and then

$$\begin{aligned} M_{57}(E=t) &= .46\cdot.1 + .54\cdot.2 = .154, \\ M_{57}(E=f) &= .46\cdot.9 + .54\cdot.8 = .846. \end{aligned}$$

Query 2: $\Pr(G, F = \text{true})$, evidence $F = \text{true}$, root 7. The same six messages serve; M_{13}, M_{23} are reused (Part 3) and the other four recomputed with λ_F the $F = \text{true}$ indicator and $\lambda_G \equiv 1$:

$$\begin{aligned} M_{64}(D) &= \Theta_{F=t|D} = (.7,\ .6), \\ M_{43}(C=t) &= .2\cdot.7 + .8\cdot.6 = .62, \\ M_{43}(C=f) &= .7\cdot.7 + .3\cdot.6 = .67, \\ M_{35}(C=t) &= .46\cdot.62 = .2852, \\ M_{35}(C=f) &= .54\cdot.67 = .3618, \\ M_{57}(E=t) &= .1\cdot.2852 + .2\cdot.3618 = .10088, \\ M_{57}(E=f) &= .9\cdot.2852 + .8\cdot.3618 = .54612. \end{aligned}$$

Query 3: evidence $A = \text{true}, F = \text{true}$, root 6. Only M_{23} survives (Part 3), so the messages toward 6 are

$$\begin{aligned} M_{13}(A) &= f_A\lambda_A = (.6,\ 0), \\ M_{75}(E) &= \textstyle\sum_G f_G\lambda_G = (1,1), \\ M_{53}(C) &= \textstyle\sum_E f_E\lambda_E M_{75} = (1,1), \\ M_{34}(C) &= \textstyle\sum_{A,B} f_C\lambda_C M_{13}M_{23}M_{53} = \Pr(C, A=t), \\ M_{46}(D) &= \textstyle\sum_C f_D\lambda_D M_{34}. \end{aligned}$$

Numerically,

$$\begin{aligned} M_{34}(C=t) &= .6\,(.5\cdot.9 + .5\cdot.1) = .30, \\ M_{34}(C=f) &= .6\,(.5\cdot.1 + .5\cdot.9) = .30, \\ M_{46}(D=t) &= .2\cdot.30 + .7\cdot.30 = .27, \\ M_{46}(D=f) &= .8\cdot.30 + .3\cdot.30 = .33. \end{aligned}$$

Part 3. A change of evidence at a node invalidates every message directed away from it (the invalidation rule of Algorithm 12, Figure 7.10).
(i) Query 1 to Query 2: evidence changes at node 6 (λ_F) and node 7 (λ_G). Away from 6 lie $M_{64}, M_{43}, M_{35}, M_{57}$ (plus M_{31}, M_{32}, M_{34}, never computed); away from 7 lie $M_{75}, M_{53}, M_{31}, M_{32}, M_{34}, M_{46}$, none computed. So exactly M_{13}, M_{23} survive.
(ii) Query 2 to Query 3: only λ_A changes, at node 1. Away from 1 lie $M_{13}, M_{32}, M_{34}, M_{35}, M_{46}, M_{57}$, which invalidates the stored M_{13}, M_{35}, M_{57}. M_{23}, M_{64}, M_{43} stay valid, but with root 6 only M_{23} points the right way; M_{64} and M_{43} point at node 3, while Query 3 needs their reverses M_{34}, M_{46}.
Part 4. Line 9 of Algorithm 12 returns $\Pr(\mathbf{C}_i, \mathbf{e}) = \phi_i \prod_k M_{ki}$ at the root, where all incoming messages are available.
Query 1. At node 7, $f_G\lambda_G M_{57}$ vanishes for $G = \text{false}$ and otherwise

$$\begin{aligned} \Pr(E=t, G=t) &= .2\cdot.154 = .0308, \\ \Pr(E=f, G=t) &= .8\cdot.846 = .6768. \end{aligned}$$

Summing out E,

$$\Pr(G = \text{true}) = .0308 + .6768 = .7076.$$

Query 2. At node 7, $f_G M_{57}$ gives

$$\Pr(E=t,G=t,F=t) = .2\cdot.10088 = .020176,$$
$$\Pr(E=f,G=t,F=t) = .8\cdot.54612 = .436896,$$
$$\Pr(E=t,G=f,F=t) = .8\cdot.10088 = .080704,$$
$$\Pr(E=f,G=f,F=t) = .2\cdot.54612 = .109224.$$

Summing out E,

$$\Pr(G=\text{true},F=\text{true}) = .457072, \qquad \Pr(G=\text{false},F=\text{true}) = .189928.$$

Query 3. At node 6, $f_F \lambda_F M_{46}$ vanishes for $F=\text{false}$ and otherwise

$$\Pr(D=t,F=t,A=t) = .7\cdot.27 = .189,$$
$$\Pr(D=f,F=t,A=t) = .6\cdot.33 = .198.$$

Summing out D,

$$\Pr(F=\text{true},A=\text{true},F=\text{true}) = .387, \qquad \Pr(F=\text{false},A=\text{true},F=\text{true}) = 0.$$

The printed third query lists F both as query variable and as evidence, so its literal answer is degenerate; on the intended reading $\Pr(F, A=\text{true})$ the same messages apply with $\lambda_F \equiv 1$ at node 6, giving

$$\Pr(F=\text{true},A=\text{true}) = .7\cdot.27 + .6\cdot.33 = .387,$$
$$\Pr(F=\text{false},A=\text{true}) = .3\cdot.27 + .4\cdot.33 = .213.$$

Problem 7.4. Prove Equation 7.1. Hint: Prove by induction, assuming first that node i has only a single neighbor and then prove for an arbitrary node i.
Equation 7.1 refers to Line 4 of Algorithm 10, FE2. There, a node $i \neq r$ with a single remaining neighbor j is removed from the current tree, $\mathbf{V}$ is the set of variables appearing in the current factor ϕ_i but not in the remaining tree, and the equation asserts

$$\sum_{\mathbf{V}} \phi_i = \text{project}(\phi_i, \mathbf{S}_{ij}),$$

that is, the factor obtained by summing out $\mathbf{V}$ from ϕ_i is a factor over the separator $\mathbf{S}_{ij}$ of Definition 7.3.

Solution. It suffices to prove that at each elimination step $\text{vars}(\phi_i^*) = \mathbf{C}_i$ and $\mathbf{V} = \mathbf{C}_i \setminus \mathbf{S}_{ij}$, since then summing out $\mathbf{V}$ is summing out $\text{vars}(\phi_i^*) \setminus \mathbf{S}_{ij}$, which is Equation 7.1.
Write T' for the current tree and ϕ_k^* for the current factor at node k, and let D_k be the neighbors of k in T already eliminated; separators, clusters and the sets $\text{vars}(i,j)$ always refer to the original T. Line 3 defines $\mathbf{V} = \text{vars}(\phi_i^*) \setminus \text{vars}(T' - i)$. Induct on the number of steps, carrying the invariant
(a) $\text{vars}(\phi_k^*) = \text{vars}(k) \cup \bigcup_{m \in D_k} \mathbf{S}_{km}$ for every $k \in T'$,
true initially since $D_k = \emptyset$ and $\text{vars}(\phi_k) = \text{vars}(k)$ (Definition 7.2). Let FE2 remove i with single remaining neighbor j. Since Line 2 removes only leaves of T', T' is always a connected subtree of T, so every neighbor of i other than j has been eliminated, i.e. D_i is all of them; (a) and Exercise 7.5 then give

$$\text{vars}(\phi_i^*) = \text{vars}(i) \cup \bigcup_{m \neq j} \mathbf{S}_{im} = \mathbf{C}_i.$$

It remains to show $\mathbf{C}_i \cap \text{vars}(T' - i) = \mathbf{S}_{ij}$. Note first that $T' - i$ is connected, avoids i, and contains j, so it lies on the j-side of the edge $i - j$.
(i) Left to right. Let $X \in \mathbf{C}_i \cap \text{vars}(T' - i)$; then $X \in \text{vars}(i,j)$ by Exercise 7.7. Also $X \in \text{vars}(\phi_k^*)$ for some remaining $k \neq i$, so by (a) either $X \in \text{vars}(k)$ or $X \in \mathbf{S}_{km} \subseteq \text{vars}(m,k)$ for some $m \in D_k$.

Both k and m lie on the j-side (m is a neighbor of k other than the uneliminated i), hence so does the whole m-side of $k-m$; thus $X \in \text{vars}(j,i)$ and $X \in \mathbf{S}_{ij}$.
(ii) Right to left. Let $X \in \mathbf{S}_{ij}$, so $X \in \mathbf{C}_i$ by Definition 7.4 and $X \in \text{vars}(p)$ for some p on the j-side. If $p \in T'$ then $X \in \text{vars}(\phi_p^*)$ by (a) and we are done. Otherwise let k be the first node of T' on the path from p to i (it exists as $j \in T'$, and $k \neq i$ as j precedes i), and let m be its predecessor, so $m \in D_k$ and $X \in \text{vars}(p) \subseteq \text{vars}(m,k)$. Since $X \in \text{vars}(i,j)$ there is s on the i-side with $X \in \text{vars}(s)$, and the path $s,\ldots,i,j,\ldots,k,m$ places s on the k-side of $k-m$, so $X \in \text{vars}(k,m)$. Hence $X \in \mathbf{S}_{km} \subseteq \text{vars}(\phi_k^*) \subseteq \text{vars}(T'-i)$.
Therefore $\mathbf{V} = \mathbf{C}_i \setminus \mathbf{S}_{ij}$ and

$$\sum_{\mathbf{V}} \phi_i^* = \sum_{\text{vars}(\phi_i^*)\setminus\mathbf{S}_{ij}} \phi_i^* = \text{project}(\phi_i^*, \mathbf{S}_{ij}),$$

which is Equation 7.1. Line 4 then multiplies this factor over $\mathbf{S}_{ij} = \mathbf{S}_{ji}$ into ϕ_j^* and changes nothing else, so (a) holds for j with the enlarged set $D_j \cup \{i\}$ and the induction closes.

Problem 7.5. Let i be a node in an elimination tree. Show that for any particular neighbor k of i, we have

$$\mathbf{S}_{ik} \subseteq \text{vars}(i) \cup \bigcup_{j\neq k} \mathbf{S}_{ij}$$

and hence

$$\mathbf{C}_i = \text{vars}(i) \cup \bigcup_{j\neq k} \mathbf{S}_{ij},$$

where the unions range over the neighbors $j \neq k$ of node i.

Solution. Both claims follow from two facts about the tree, with j ranging over the neighbors of i other than k and $\mathbf{S}_{ij} = \text{vars}(i,j) \cap \text{vars}(j,i)$ as in Definition 7.3.
Fact 1. Deleting $i-k$ leaves the component of i, which is $\{i\}$ together with the subtrees hanging off i through its other neighbors, so

$$\text{vars}(i,k) = \text{vars}(i) \cup \bigcup_{j\neq k} \text{vars}(j,i).$$

Fact 2. Every node on the k-side of $i-k$ reaches j by a path through k and then i, so deleting $i-j$ leaves it in the component of i; hence $\text{vars}(k,i) \subseteq \text{vars}(i,j)$.
Let $X \in \mathbf{S}_{ik}$, so $X \in \text{vars}(i,k)$ and $X \in \text{vars}(k,i)$. By Fact 1 either $X \in \text{vars}(i)$, or $X \in \text{vars}(j,i)$ for some $j \neq k$, in which case Fact 2 gives $X \in \text{vars}(k,i) \subseteq \text{vars}(i,j)$ and hence $X \in \mathbf{S}_{ij}$. Either way X lies in $\text{vars}(i) \cup \bigcup_{j\neq k} \mathbf{S}_{ij}$, which is the first claim. (If k is the only neighbor of i, the second alternative cannot arise and $\mathbf{S}_{ik} \subseteq \text{vars}(i,k) = \text{vars}(i)$.)
Definition 7.4 then gives

$$\mathbf{C}_i = \Big(\text{vars}(i) \cup \bigcup_{j\neq k} \mathbf{S}_{ij}\Big) \cup \mathbf{S}_{ik} = \text{vars}(i) \cup \bigcup_{j\neq k} \mathbf{S}_{ij},$$

the trailing term being absorbed by the first claim.

Problem 7.6. Prove the following statement with respect to Algorithm 12, FE:

$$M_{ij} = \text{project}(\phi_{ij}, \mathbf{S}_{ij}),$$

where ϕ_{ij} is the product of all factors assigned to nodes on the i-side of edge $i-j$ in the elimination tree. Use this result to show that

$$M_{ij}M_{ji} = \Pr(\mathbf{S}_{ij}, \mathbf{e})$$

for every edge $i-j$ in the elimination tree.

Solution. Part 1 is an induction on the number of nodes on the i-side of $i-j$, with ϕ_{ij} the product of all factors (CPTs and indicators, by Lines 1 to 5 of Algorithm 12) assigned there, so that $\text{vars}(\phi_{ij}) = \text{vars}(i,j)$. Throughout, a factor pulls out of any summation over variables it does not mention, and summations over distinct variables commute.
Base case: the i-side is $\{i\}$, so i is a leaf, the product in Equation 7.2 is empty and $\phi_{ij} = \phi_i$.
Inductive step: let $k_1, \ldots, k_n$ be the neighbors of i other than j. Deleting i splits the i-side into $\{i\}$ and the pairwise disjoint k_t-sides of the edges $k_t - i$ (Fact 1 of Exercise 7.5), so

$$\phi_{ij} = \phi_i \prod_{t=1}^{n} \phi_{k_t i}, \qquad \text{vars}(\phi_{ij}) = \text{vars}(i) \cup \bigcup_t \text{vars}(k_t, i),$$

and each k_t-side is smaller, so $M_{k_t i} = \text{project}(\phi_{k_t i}, \mathbf{S}_{k_t i})$ by hypothesis. Put $\mathbf{U}_t = \text{vars}(k_t, i) \setminus \mathbf{S}_{k_t i}$. A variable of $\mathbf{U}_t$ lies in $\text{vars}(k_t, i)$ but not $\text{vars}(i, k_t)$, hence occurs in no factor outside the k_t-side; so the $\mathbf{U}_t$ are pairwise disjoint and disjoint from $\text{vars}(i)$, from every $\mathbf{S}_{k_s i}$, and from $\mathbf{S}_{ij} \subseteq \text{vars}(j, i)$. They are therefore among the variables that $\text{project}(\phi_{ij}, \mathbf{S}_{ij})$ sums out, and may be summed out first:

$$\begin{aligned} \sum_{\mathbf{U}_1 \cup \cdots \cup \mathbf{U}_n} \phi_i \prod_t \phi_{k_t i} &= \phi_i \prod_t \Big(\sum_{\mathbf{U}_t} \phi_{k_t i} \Big) \\ &= \phi_i \prod_t \text{project}(\phi_{k_t i}, \mathbf{S}_{k_t i}) = \phi_i \prod_t M_{k_t i}, \end{aligned}$$

the second equality using $\text{vars}(\phi_{k_t i}) \setminus \mathbf{S}_{k_t i} = \mathbf{U}_t$. The variables so removed are exactly $\bigcup_t \mathbf{U}_t$, and by the disjointness above

$$\text{vars}(\phi_{ij}) \setminus \bigcup_t \mathbf{U}_t = \text{vars}(i) \cup \bigcup_t \mathbf{S}_{k_t i} = \text{vars}\Big(\phi_i \prod_t M_{k_t i}\Big),$$

which is the cluster $\mathbf{C}_i \supseteq \mathbf{S}_{ij}$ by Exercise 7.5. Summing out what remains, $\mathbf{C}_i \setminus \mathbf{S}_{ij}$, and applying Equation 7.2,

$$\text{project}(\phi_{ij}, \mathbf{S}_{ij}) = \text{project}\Big(\phi_i \prod_t M_{k_t i}, \mathbf{S}_{ij}\Big) = M_{ij}.$$

Part 2. By Part 1 and $\mathbf{S}_{ji} = \mathbf{S}_{ij}$, both M_{ij} and M_{ji} are factors over $\mathbf{S}_{ij}$. Write $\mathbf{A} = \text{vars}(i,j) \setminus \mathbf{S}_{ij}$ and $\mathbf{B} = \text{vars}(j,i) \setminus \mathbf{S}_{ij}$; Definition 7.3 makes $\mathbf{A}, \mathbf{B}$ disjoint, with no variable of $\mathbf{B}$ in ϕ_{ij} and none of $\mathbf{A}$ in ϕ_{ji}, so

$$\begin{aligned} \text{project}(\phi_{ij}\phi_{ji}, \mathbf{S}_{ij}) &= \sum_{\mathbf{A}} \sum_{\mathbf{B}} \phi_{ij}\phi_{ji} = \sum_{\mathbf{A}} \phi_{ij} \sum_{\mathbf{B}} \phi_{ji} \\ &= \Big(\sum_{\mathbf{A}} \phi_{ij} \Big) \Big(\sum_{\mathbf{B}} \phi_{ji} \Big) = M_{ij} M_{ji}. \end{aligned}$$

Every node lies on exactly one side of $i-j$ and every CPT and indicator is assigned to exactly one node (Definition 7.2, Lines 1 to 5 of Algorithm 12), so $\phi_{ij}\phi_{ji}$ is the product of all factors,

$$\phi_{ij}\phi_{ji} = \prod_X \Theta_{X|\mathbf{P}_X} \prod_{E \in \mathbf{e}} \lambda_E,$$

with $\mathbf{P}_X$ the parents of X. By the chain rule (Equation 4.2) the first product evaluates at an instantiation $\mathbf{z}$ of all network variables to $\Pr(\mathbf{z})$, while the second is 1 if $\mathbf{z} \sim \mathbf{e}$ and 0 otherwise. Hence for each instantiation $\mathbf{s}$ of $\mathbf{S}_{ij}$,

$$\text{project}(\phi_{ij}\phi_{ji}, \mathbf{S}_{ij})(\mathbf{s}) = \sum_{\mathbf{z} \sim \mathbf{s}} \Pr(\mathbf{z}) \prod_E \lambda_E(\mathbf{z}) = \sum_{\mathbf{z} \sim \mathbf{s},\ \mathbf{z} \sim \mathbf{e}} \Pr(\mathbf{z}) = \Pr(\mathbf{s}, \mathbf{e}).$$

Combining the last two displays, $M_{ij}M_{ji} = \Pr(\mathbf{S}_{ij}, \mathbf{e})$.

Problem 7.7. Prove that for every edge $i - j$ in an elimination tree, $\mathbf{S}_{ij} = \mathbf{C}_i \cap \mathbf{C}_j$.

Solution. One inclusion is immediate from Definition 7.4: the union defining $\mathbf{C}_i$ has j among its neighbors and so contains the term $\mathbf{S}_{ij}$, while the union defining $\mathbf{C}_j$ contains $\mathbf{S}_{ji} = \mathbf{S}_{ij}$ (the expression of Definition 7.3 is symmetric). Hence $\mathbf{S}_{ij} \subseteq \mathbf{C}_i \cap \mathbf{C}_j$.
The reverse inclusion follows from the claim that $\mathbf{C}_i \subseteq \text{vars}(i, j)$ for every edge $i - j$. Since $\mathbf{C}_i = \text{vars}(i) \cup \bigcup_m \mathbf{S}_{im}$, check the three kinds of piece: $\text{vars}(i) \subseteq \text{vars}(i, j)$ as i sits on the i-side; $\mathbf{S}_{ij} \subseteq \text{vars}(i, j)$ by Definition 7.3; and for a neighbor $m \neq j$, every node on the m-side of $i - m$ reaches j through m and then i, so deleting $i - j$ leaves it in the component of i, giving $\mathbf{S}_{im} \subseteq \text{vars}(m, i) \subseteq \text{vars}(i, j)$.
Applying the claim at both ends of the edge,

$$\mathbf{C}_i \cap \mathbf{C}_j \subseteq \text{vars}(i, j) \cap \text{vars}(j, i) = \mathbf{S}_{ij}.$$

6.2 Exercises 7.8–7.14

Problem 7.8. Let N be a Bayesian network and let (T, ϕ) be an elimination tree of width w for the CPTs of network N. Show how to construct an elimination order for network N of width $\leq w$.
Hint: Consider the order in which variables are eliminated in the context of factor elimination.

Solution. Run FE3 (Algorithm 11) on (T, ϕ) from any root r, let $i_1, \ldots, i_{m-1}$ be the nodes it removes and j_k the unique remaining neighbour of i_k at step k, and take

$$\pi \;=\; V_1,\; \ldots,\; V_{m-1},\; C_r, \qquad V_k \;=\; C_{i_k} \setminus S_{i_k j_k},$$

the variables inside each block in arbitrary order. This is well defined: i_k is eliminated only once all its neighbours but j_k are gone, so by Exercise 7.5 its current factor is over exactly C_{i_k}, and projecting on $S_{i_k j_k}$ sums out V_k.
π is a legal order. By Line 3 of FE2 a variable $X \in V_k$ appears in ϕ_{i_k} but in no factor of the tree remaining after step k, so it lies in no later block and not in C_r; conversely every network variable occurs in some CPT, hence is either summed out at a unique step or survives into ϕ_r, i.e. into C_r. So π lists every variable exactly once.
Write T_k for the tree remaining just before step k. Two observations:
(1) A variable $X \in V_k$ occurs only in the factor ϕ_{i_k} of T_k, hence in no separator of T_k (a separator needs factors on both sides of its edge) and, since $C_l = \text{vars}(l) \cup \bigcup_m S_{lm}$, in no cluster of T_k other than C_{i_k}.
(2) The clusters and separators of T_k are the original ones throughout. Passing to T_{k+1} deletes exactly V_k from one side of each edge, and by (1) those variables occur nowhere else, so no separator changes; the factor at j_k gains $S_{i_k j_k}$, exactly the separator term its cluster loses with the deleted edge, and no other factor changes.
Width. On the interaction graph, eliminating X connects its current neighbours pairwise and deletes X, and the width is the largest $\{X\}$ together with those neighbours, minus one. Let G_k be the interaction graph just before block V_k, and H_k the graph whose edges are the pairs lying together in some cluster C_l, $l \in T_k$. Then $G_k \subseteq H_k$: for $k = 1$ each CPT sits at a node i with its variables inside $\text{vars}(i) \subseteq C_i$; and given $G_k \subseteq H_k$, each $X \in V_k$ lies by (1) in the single cluster C_{i_k}, so its G_k-neighbours lie in $C_{i_k} \setminus \{X\}$, its cluster is contained in C_{i_k}, and the fill-in edges it creates join variables already adjacent in H_k. After the block the surviving cliques are the C_l, $l \neq i_k$, together with $C_{i_k} \setminus V_k = S_{i_k j_k} \subseteq C_{j_k}$, which by (2) are the cliques of H_{k+1}.
After step $m - 1$ only r survives, the remaining variables are C_r, and the graph is a subgraph of that clique. Every variable of π therefore has its cluster inside some C_i, so

$$\text{width}(\pi) \;\leq\; \max_i |C_i| - 1 \;=\; w.$$

Problem 7.9. Consider the following set of factors: $f(ABE)$, $f(ACD)$, and $f(DEF)$. Show that the optimal elimination tree (the one with smallest width) for this set of factors must have more than three nodes (hence, the set of nodes in the elimination tree cannot be in one-to-one correspondence with the factors).

Solution. The optimal width is 2, attained by the four-node star below and by no tree with three or fewer nodes.
Lower bound 2: each factor sits at some node i with $\text{vars}(i) \subseteq C_i$, and each has three variables, so some cluster has size ≥ 3.
Width 2 with four nodes. Take the star T with a fourth node 4 carrying no factor, adjacent to the three factor nodes $1, 2, 3$:

$$\phi_1 = f(ABE), \quad \phi_2 = f(ACD), \quad \phi_3 = f(DEF), \quad \phi_4 = 1,$$

with edges $1-4$, $2-4$, $3-4$ (Definition 7.2 requires only that each factor sit at exactly one node, so node 4 may carry the empty product, $\text{vars}(4) = \emptyset$). The separators are

$$\begin{aligned} S_{14} &= \{A, B, E\} \cap \{A, C, D, E, F\} = \{A, E\}, \\ S_{24} &= \{A, C, D\} \cap \{A, B, D, E, F\} = \{A, D\}, \\ S_{34} &= \{D, E, F\} \cap \{A, B, C, D, E\} = \{D, E\}, \end{aligned}$$

and therefore the clusters are

$$\begin{aligned} C_1 &= \{A, B, E\}, \quad C_2 = \{A, C, D\}, \\ C_3 &= \{D, E, F\}, \quad C_4 = \{A, D, E\}. \end{aligned}$$

All four clusters have size 3, so the width is 2, optimal by the lower bound.
At most three nodes. Two cases.
(i) Some node carries two factors. Its cluster contains the union of their variables, and

$$ABE \cup ACD = ABCDE, \quad ABE \cup DEF = ABDEF, \quad ACD \cup DEF = ACDEF$$

each have size 5, so the width is at least 4.
(ii) Exactly three nodes, one factor each — the one-to-one case. A tree on three nodes is a path, and in each of the three arrangements the middle node absorbs both separators into a cluster of size 4, giving width 3. Writing $1, 2, 3$ for the nodes of $f(ABE)$, $f(ACD)$, $f(DEF)$:

- $1-2-3$: $S_{12} = \{A, E\}$, $S_{23} = \{D, E\}$, so $C_2 = \{A, C, D, E\}$.
- $2-1-3$: $S_{12} = \{A, D\}$, $S_{13} = \{D, E\}$, so $C_1 = \{A, B, D, E\}$.
- $1-3-2$: $S_{13} = \{A, E\}$, $S_{23} = \{A, D\}$, so $C_3 = \{A, D, E, F\}$.

So no elimination tree with three or fewer nodes attains width 2, and the optimal tree has more than three nodes.

Problem 7.10. A cutset $\mathbf{C}$ for a DAG G is a set of nodes that renders G a polytree when all edges outgoing from nodes $\mathbf{C}$ are removed. Let k be the maximum number of parents per node in the DAG G. Show how to construct an elimination tree for G whose width is $\leq k + |\mathbf{C}|$.

Solution. Let G' be G with every edge leaving $\mathbf{C}$ deleted, a polytree by hypothesis, and take T with one node per variable of G, edges those of the skeleton of G' plus any set joining its components into a single tree, and node X carrying the original CPT $\Theta_{X|\mathbf{U}_X}$, where $\mathbf{U}_X$ are the parents of X in G. By Definition 7.2 this is an elimination tree (each CPT sits at exactly one node), $\text{vars}(X) = \{X\} \cup \mathbf{U}_X$ has at most $k + 1$ variables, and its width is at most $k + |\mathbf{C}|$.
Key observation: an edge $Y \to Z$ of G survives in G' unless $Y \in \mathbf{C}$, so every parent of Z is either in $\mathbf{C}$ or a skeleton neighbour of Z, hence a neighbour of Z in T.
Separators. Let $X-Y$ be an edge of T, splitting it into $T_X \ni X$ and $T_Y \ni Y$, and take $Z \in T_X$ with a parent $W \in \mathbf{U}_Z \setminus \mathbf{C}$. By the observation $Z-W$ is an edge of T, and the only edge crossing the cut is

$X-Y$, so either $W \in T_X$ or $Z = X$ and $W = Y$; hence $\mathrm{vars}(X,Y) \subseteq T_X \cup \mathbf{C} \cup \{Y\}$, and symmetrically $\mathrm{vars}(Y,X) \subseteq T_Y \cup \mathbf{C} \cup \{X\}$. If $X-Y$ comes from the polytree edge $X \to Y$ the exception cannot arise, since it would put $Y \in \mathbf{U}_X$ against the acyclicity of G'; then $\mathrm{vars}(X,Y) \subseteq T_X \cup \mathbf{C}$ and, as $T_X \cap T_Y = \emptyset$,

$$S_{XY} \;\subseteq\; \big(T_X \cup \mathbf{C}\big) \cap \big(T_Y \cup \mathbf{C} \cup \{X\}\big) \;=\; \{X\} \cup \mathbf{C}.$$

If $X-Y$ is one of the added edges, no skeleton edge crosses the cut and the same computation gives $S_{XY} \subseteq \mathbf{C}$.

Clusters. A neighbour Y of X in T is of one of three kinds: $X \to Y$ in G', giving $S_{XY} \subseteq \{X\} \cup \mathbf{C}$; $Y \to X$ in G', giving $S_{XY} \subseteq \{Y\} \cup \mathbf{C}$ by the same argument with the roles swapped, and $Y \in \mathbf{U}_X$ since Y is then a parent of X in G' hence in G; or joined by an added edge, giving $S_{XY} \subseteq \mathbf{C}$. In every case $S_{XY} \subseteq \{X\} \cup \mathbf{U}_X \cup \mathbf{C}$, so

$$C_X \;\subseteq\; \{X\} \cup \mathbf{U}_X \cup \mathbf{C}, \qquad \max_X |C_X| - 1 \;\le\; k + |\mathbf{C}|.$$

Problem 7.11. Show that the message defined by Equation 7.4,

$$M_{ij} \;=\; \sum_{C_i \setminus S_{ij}} \Phi_i \prod_{k \neq j} M_{ki},$$

can be computed in space that is exponential only in the size of separator S_{ij}, given messages M_{ki}, $k \neq j$, and the factors assigned to node i (here Φ_i is the product of the factors, including evidence indicators, assigned to node i).

Solution. Accumulate the sum one cluster instantiation at a time, so that $F = \Phi_i \prod_{k \neq j} M_{ki}$ — the factor over C_i that a naive reading of Equation 7.4 would build first — is never materialised. Let $f_1, \ldots, f_p$ be the factors at node i (CPTs and indicators) and $M_{k_1 i}, \ldots, M_{k_q i}$ the incoming messages, $k_s \neq j$; all are already-stored inputs.

1. Allocate M_{ij} over S_{ij} and set $M_{ij}(\mathbf{s}) \leftarrow 0$ for every instantiation $\mathbf{s}$.
2. Enumerate the instantiations $\mathbf{c}$ of C_i one at a time, by an odometer over its variables. For each, compute the single number $v = \prod_t f_t(\mathbf{c}_{\mathrm{vars}(f_t)}) \prod_s M_{k_s i}(\mathbf{c}_{S_{k_s i}})$, writing $\mathbf{c}_{\mathbf{X}}$ for the restriction of $\mathbf{c}$ to $\mathbf{X}$, and update $M_{ij}(\mathbf{s}) \leftarrow M_{ij}(\mathbf{s}) + v$ at $\mathbf{s} = \mathbf{c}_{S_{ij}}$.
3. Return M_{ij}.

Correctness: every factor in the product is over a subset of C_i, since $\mathrm{vars}(f_t) \subseteq \mathrm{vars}(i) \subseteq C_i$ and $S_{k_s i} \subseteq C_i$ by Definition 7.4, so $\mathbf{c}$ determines each of their values and $v = F(\mathbf{c})$. Grouping the $\mathbf{c}$ by their restriction to S_{ij}, the accumulator holds

$$M_{ij}(\mathbf{s}) \;=\; \sum_{\mathbf{c} \sim \mathbf{s}} F(\mathbf{c}) \;=\; \Big(\sum_{C_i \setminus S_{ij}} F\Big)(\mathbf{s}),$$

which is Equation 7.4.

Space: beyond its inputs the procedure allocates only the output table, of $\prod_{X \in S_{ij}} |X|$ entries, and the odometer, running product and indices, $O(|C_i|)$ numbers — in total $O(\exp(|S_{ij}|) + |C_i|)$, exponential in the separator alone. Nor are the inputs exponential in $|C_i|$: each f_t is a CPT or an evidence indicator, and each $M_{k_s i}$ is stored over a separator.

Problem 7.12. Consider the elimination tree on the left of Figure 7.13. The underlying network is the DAG of Figure 7.12(a), with edges

$$\begin{aligned} &A \to B, \quad A \to C, \quad A \to D, \quad A \to E, \quad A \to F, \\ &B \to D, \quad C \to E, \quad D \to G, \quad F \to G, \quad E \to H, \quad F \to H, \end{aligned}$$

so that its CPTs are Θ_A, $\Theta_{B|A}$, $\Theta_{C|A}$, $\Theta_{D|AB}$, $\Theta_{E|AC}$, $\Theta_{F|A}$, $\Theta_{G|DF}$, $\Theta_{H|EF}$. The elimination tree in question has six nodes with edges

$$1-3, \qquad 3-4, \qquad 3-5, \qquad 2-4, \qquad 4-6,$$

and factor assignment

node	assigned factors	vars
1	$\Theta_{G\|DF}$	DFG
2	$\Theta_{C\|A}\Theta_{E\|AC}$	ACE
3	none (trivial factor)	$\emptyset$
4	$\Theta_A\Theta_{F\|A}$	AF
5	$\Theta_{B\|A}\Theta_{D\|AB}$	ABD
6	$\Theta_{H\|EF}$	EFH

Its separators are $S_{13} = DF$, $S_{34} = AF$, $S_{35} = AD$, $S_{24} = AE$, $S_{46} = EF$, and its clusters are those of the minimal jointree of Figure 7.12(b):

$$C_1 = DFG,\ C_2 = ACE,\ C_3 = ADF,\ C_4 = AEF,\ C_5 = ABD,\ C_6 = EFH,$$

so the tree has width 2.

Using factor elimination on this tree, one cannot compute the marginal over $AEFH$ as these variables are not contained in any cluster. Construct an elimination tree with width 3 that allows us to compute the marginal over these variables. Construct another elimination tree with width 3 that allows us to compute the marginal over variables GH.

Note: You may need to introduce auxiliary factors.

Solution. Part 1. Add the auxiliary factor $f(H) \equiv 1$ at node 4, keeping the printed structure and all other assignments; a factor identically one changes no marginal and only enlarges vars(4) to $\{A, F, H\}$, which is the elimination-tree form of the nonminimal-jointree trick of Section 7.7.2. Since FE3 (Algorithm 11) needs a root r with $\mathbf{Q} \subseteq C_r$, it is enough to drive H into C_4; the separators carry A, E, F there already.

Only S_{46} changes: the 6-side of $4-6$ has variables $\{E, F, H\}$ and the 4-side now has $\{D, F, G\} \cup \{A, C, E\} \cup \{A, F, H\} \cup \{A, B, D\}$, all eight, so

$$S_{46} = \{E, F, H\}.$$

The rest are as before: on 3–4 the two sides are $\{A, C, E, F, H\}$ and $\{A, B, D, F, G\}$, giving $S_{34} = AF$; likewise $S_{13} = DF$, $S_{35} = AD$, $S_{24} = AE$. Hence

$$C_1 = DFG, \quad C_2 = ACE, \quad C_3 = ADF,$$
$$C_4 = AEFH, \quad C_5 = ABD, \quad C_6 = EFH.$$

since $C_4 = \{A, F, H\} \cup \{A, F\} \cup \{A, E\} \cup \{E, F, H\}$. The largest cluster has four variables, so the width is 3, and with root 4, FE3 returns $\Pr(AEFH)$.

Part 2. Add $f(GH) \equiv 1$ at node 1, so vars(1) $= \{D, F, G, H\}$, all other assignments unchanged. A one-variable factor will not do here: G occurs only at node 1 and H only at node 6, three edges away, so H has to be carried along the path $6 - 4 - 3 - 1$. Along it H now appears on both sides of every edge:

$$S_{46} = \{E, F, H\} \cap \{A, B, C, D, E, F, G, H\} = \{E, F, H\},$$
$$S_{34} = \{A, B, D, F, G, H\} \cap \{A, C, E, F, H\} = \{A, F, H\},$$
$$S_{13} = \{D, F, G, H\} \cap \{A, B, C, D, E, F, H\} = \{D, F, H\},$$

while $S_{35} = AD$ and $S_{24} = AE$ are untouched. The clusters are

$$C_1 = DFGH, \quad C_2 = ACE, \quad C_3 = ADFH,$$
$$C_4 = AEFH, \quad C_5 = ABD, \quad C_6 = EFH.$$

for instance $C_3 = S_{13} \cup S_{34} \cup S_{35} = \{D, F, H\} \cup \{A, F, H\} \cup \{A, D\}$. The largest clusters have size 4, so the width is again 3; with root 1, FE3 returns $\Pr(C_1) = \Pr(DFGH)$, whose projection on GH is $\Pr(GH)$.

Problem 7.13. Consider the polytree algorithm as discussed in Section 7.5.4. Here the Bayesian network has a polytree structure, and the elimination tree used is the one of Figure 7.11: it has one node per network variable X, its edges are the edges of the polytree (with directions dropped), and the CPT $\Theta_{X|\mathbf{U}_X}$ together with the evidence indicator λ_X of variable X are assigned to the corresponding node.
Let $X \to Y$ be an edge in the polytree. Show that the messages M_{XY} and M_{YX} sent across this edge must be over variable X.

Solution. Both messages are factors over the separator S_{XY}, since Equation 7.2 projects onto it and $S_{ij} = S_{ji}$; so it is enough to show $S_{XY} = \{X\}$.
Node Z carries $\Theta_{Z|\mathbf{U}_Z}$ and λ_Z, so $\text{vars}(Z) = \{Z\} \cup \mathbf{U}_Z$ is the family of Z. Removing $X-Y$ splits the tree into the disjoint $T_X \ni X$ and $T_Y \ni Y$, and by Definition 7.3, $\text{vars}(X,Y) = \bigcup_{Z \in T_X} (\{Z\} \cup \mathbf{U}_Z)$.
The X-side. Let $Z \in T_X$ and $W \in \mathbf{U}_Z$, so that $W-Z$ is an edge of the elimination tree. The only edge joining T_X to T_Y is $X-Y$, so $W \in T_X$ unless $Z = X$ and $W = Y$ — and that exception would give $Y \to X$, impossible alongside $X \to Y$ in an acyclic polytree. Hence $\text{vars}(X,Y) = T_X$, the reverse inclusion holding since $Z \in \text{vars}(Z)$.
The Y-side. The same argument permits exactly one exception, $Z = Y$ with $W = X$, and it does occur since $X \to Y$ means $X \in \mathbf{U}_Y$. Hence $\text{vars}(Y,X) = T_Y \cup \{X\}$, and

$$S_{XY} \;=\; T_X \cap \big(T_Y \cup \{X\}\big) \;=\; \{X\}.$$

Problem 7.14. Consider the polytree algorithm as discussed in Section 7.5.4 — the elimination tree has one node per network variable X, its edges are those of the polytree, and node X carries $\Theta_{X|\mathbf{U}_X}$ and the evidence indicator λ_X — and let $\mathbf{e}$ be some given evidence. Let $X \to Y$ be an edge in the polytree, let $\mathbf{e}^+_{XY}$ be the evidence assigned to nodes on the X-side of this edge, and let $\mathbf{e}^-_{XY}$ be the evidence assigned to nodes on the Y-side of the edge (hence $\mathbf{e} = \mathbf{e}^+_{XY}, \mathbf{e}^-_{XY}$). Show that the messages passed by Algorithm 12, FE, across edges $X \to Y$ have the following meaning:

$$M_{XY} = \Pr(X, \mathbf{e}^+_{XY}) \qquad \text{and} \qquad M_{YX} = \Pr(\mathbf{e}^-_{XY} \mid X).$$

Moreover, show that $M_{XY} M_{YX} = \Pr(X, \mathbf{e})$.

Solution. Write $T_X \ni X$ and $T_Y \ni Y$ for the two sides of the edge $X-Y$, and $\phi_Z = \Theta_{Z|\mathbf{U}_Z}\lambda_Z$ for the node factors, so that $\mathbf{e}^+_{XY}$ is the part of $\mathbf{e}$ on T_X and $\mathbf{e}^-_{XY}$ the part on T_Y. By Exercise 7.13,

$$S_{XY} = \{X\}, \qquad \text{vars}(X,Y) = T_X, \qquad \text{vars}(Y,X) = T_Y \cup \{X\},$$

and by Exercise 7.6, $M_{ij} = \text{project}(\phi_{ij}, S_{ij})$ with $\phi_{ij} = \prod_{Z \in T_i} \phi_Z$. Hence

$$M_{XY} \;=\; \sum_{T_X \setminus \{X\}} \prod_{Z \in T_X} \Theta_{Z|\mathbf{U}_Z} \lambda_Z,$$

a factor over X. Exercise 7.13 also shows every parent of a node of T_X lies in T_X, so T_X is ancestrally closed and

$$\prod_{Z \in T_X} \Theta_{Z|\mathbf{U}_Z} \;=\; \Pr(T_X)\,:$$

sum the chain-rule product $\prod_Z \Theta_{Z|\mathbf{U}_Z} = \Pr$ (Equation 4.2) over the complement in reverse topological order, where each variable summed has no child left in the product, so occurs only in its own CPT, which sums to 1. The remaining factor $\prod_{Z \in T_X} \lambda_Z$ is the indicator of $\mathbf{e}^+_{XY}$, so for each value x of X,

$$M_{XY}(x) \;=\; \sum_{\substack{\mathbf{t} \sim x \\ \mathbf{t} \sim \mathbf{e}^+_{XY}}} \Pr(\mathbf{t}) \;=\; \Pr(x, \mathbf{e}^+_{XY}),$$

the sum ranging over instantiations $\mathbf{t}$ of T_X. That is, $M_{XY} = \Pr(X, \mathbf{e}^+_{XY})$.
Backward, Exercise 7.6 with $i = Y$ gives $M_{YX} = \sum_{T_Y} \prod_{Z \in T_Y} \Theta_{Z|\mathbf{U}_Z} \lambda_Z$, again a factor over X. Every parent of a node of T_Y lies in $T_Y \cup \{X\}$, so $\prod_{Z \in T_Y} \Theta_{Z|\mathbf{U}_Z}$ is a function of $T_Y \cup \{X\}$ alone, and multiplying it by $\Pr(T_X)$ recovers the full chain-rule product:

$$\Pr(T_X, T_Y) \;=\; \Pr(T_X) \prod_{Z \in T_Y} \Theta_{Z|\mathbf{U}_Z}.$$

So wherever $\Pr(\mathbf{t}_X) > 0$ that product equals $\Pr(\mathbf{t}_Y \mid \mathbf{t}_X)$, and as it depends on $\mathbf{t}_X$ only through x, the common value is $\Pr(\mathbf{t}_Y \mid x)$ for every x with $\Pr(x) > 0$. (Where $\Pr(x) = 0$ the conditional is undefined and the product stands in for it; nothing is affected, since then $M_{XY}(x) = \Pr(x, \mathbf{e}^+_{XY}) = 0 = \Pr(x, \mathbf{e})$.)
Consequently

$$M_{YX}(x) \;=\; \sum_{\mathbf{t}_Y \sim \mathbf{e}^-_{XY}} \Pr(\mathbf{t}_Y \mid x) \;=\; \Pr(\mathbf{e}^-_{XY} \mid x),$$

i.e. $M_{YX} = \Pr(\mathbf{e}^-_{XY} \mid X)$.
Finally, that same factorisation, with $\mathbf{e}^+_{XY}$ constraining only T_X and $\mathbf{e}^-_{XY}$ only T_Y, gives for each x

$$\begin{aligned}
\Pr(x, \mathbf{e}) &= \sum_{\substack{\mathbf{t}_X \sim x,\, \mathbf{t}_X \sim \mathbf{e}^+_{XY} \\ \mathbf{t}_Y \sim \mathbf{e}^-_{XY}}} \Pr(\mathbf{t}_X) \Pr(\mathbf{t}_Y \mid x) \\
&= \Big(\sum_{\mathbf{t}_X \sim x,\, \mathbf{e}^+_{XY}} \Pr(\mathbf{t}_X) \Big) \Big(\sum_{\mathbf{t}_Y \sim \mathbf{e}^-_{XY}} \Pr(\mathbf{t}_Y \mid x) \Big) \\
&= \Pr(x, \mathbf{e}^+_{XY}) \; \Pr(\mathbf{e}^-_{XY} \mid x) \\
&= M_{XY}(x)\, M_{YX}(x).
\end{aligned}$$

Hence $M_{XY} M_{YX} = \Pr(X, \mathbf{e})$, which is also the second half of Exercise 7.6 at the separator $S_{XY} = \{X\}$.

6.3 Exercises 7.15–7.15

Problem 7.15. Definition 7.7 showed how we can divide two factors f_1 and f_2 when each is over the same set of variables $\mathbf{X}$: the quotient f_1/f_2 is the factor over $\mathbf{X}$ given by $f(\mathbf{x}) = f_1(\mathbf{x})/f_2(\mathbf{x})$ if $f_2(\mathbf{x}) \neq 0$, and $f(\mathbf{x}) = 0$ otherwise. Let us define division more generally while assuming that factor f_2 is over variables $\mathbf{Y} \subseteq \mathbf{X}$. The result is a factor f over variables $\mathbf{X}$ defined as follows:

$$f(\mathbf{x}) \;\stackrel{\text{def}}{=}\; \begin{cases} f_1(\mathbf{x})/f_2(\mathbf{y}) & \text{if } f_2(\mathbf{y}) \neq 0, \text{ for } \mathbf{y} \sim \mathbf{x} \\ 0 & \text{otherwise.} \end{cases}$$

Show that

$$\sum_{\mathbf{X} \setminus \mathbf{Y}} f_1/f_2 \;=\; \left(\sum_{\mathbf{X} \setminus \mathbf{Y}} f_1 \right) \Big/ f_2.$$

Use this fact to provide a different definition for the messages passed by the Hugin architecture. Recall that Hugin propagation maintains over each separator S_{ij} a single factor Φ_{ij}, each entry of which is initialized to 1, and over each cluster $\mathbf{C}_i$ a factor Φ_i, initialized to $\Psi_i \prod_j \Phi_{ij}$ where Ψ_i is the product of the factors (including evidence indicators) assigned to node i and j ranges over the neighbors of i. When node i is ready to send a message to neighbor j it saves Φ_{ij} into Φ^{old}_{ij}, computes the new separator factor $\Phi_{ij} \leftarrow \sum_{\mathbf{C}_i \setminus S_{ij}} \Phi_i$, computes the message $M_{ij} = \Phi_{ij}/\Phi^{old}_{ij}$, and multiplies it into the receiving cluster, $\Phi_j \leftarrow \Phi_j M_{ij}$.

Solution. Part 1. Both sides are factors over $\mathbf{Y}$ — on the left, f_1/f_2 is over $\mathbf{X}$ by the generalized definition and $\mathbf{Z} = \mathbf{X} \setminus \mathbf{Y}$ is summed out; on the right, $\sum_{\mathbf{Z}} f_1$ and f_2 are both over $\mathbf{Y}$, so their quotient

is the ordinary one of Definition 7.7. Fix an instantiation $\mathbf{y}$. The instantiations summed on the left are exactly the $\mathbf{x} = \mathbf{yz}$, all with the same compatible $\mathbf{y}$, so both the divisor $f_2(\mathbf{y})$ and the applicable case of the definition are constant across the sum.
(i) $f_2(\mathbf{y}) \neq 0$. The first case applies at every $\mathbf{yz}$, so

$$\left(\sum_{\mathbf{Z}} f_1/f_2\right)(\mathbf{y}) = \sum_{\mathbf{z}} (f_1/f_2)(\mathbf{yz}) = \sum_{\mathbf{z}} \frac{f_1(\mathbf{yz})}{f_2(\mathbf{y})}$$
$$= \frac{1}{f_2(\mathbf{y})} \sum_{\mathbf{z}} f_1(\mathbf{yz}) = \frac{\left(\sum_{\mathbf{Z}} f_1\right)(\mathbf{y})}{f_2(\mathbf{y})}.$$

the pull-out being legitimate exactly because f_2 mentions no variable of $\mathbf{Z}$; and since $f_2(\mathbf{y}) \neq 0$ this is the value of $(\sum_{\mathbf{Z}} f_1)/f_2$ at $\mathbf{y}$ under Definition 7.7.
(ii) $f_2(\mathbf{y}) = 0$. The second case applies at every $\mathbf{yz}$, so

$$\left(\sum_{\mathbf{Z}} f_1/f_2\right)(\mathbf{y}) = \sum_{\mathbf{z}} 0 = 0.$$

On the other side, the divisor f_2 vanishes at $\mathbf{y}$, so Definition 7.7 gives $\left(\left(\sum_{\mathbf{Z}} f_1\right)/f_2\right)(\mathbf{y}) = 0$ as well. The two sides agree at every $\mathbf{y}$, hence

$$\sum_{\mathbf{X}\setminus\mathbf{Y}} f_1/f_2 = \left(\sum_{\mathbf{X}\setminus\mathbf{Y}} f_1\right) \Big/ f_2,$$

Part 2. At the moment node i is ready to send to j, the identity applies with $\mathbf{X} = \mathbf{C}_i$, $\mathbf{Y} = S_{ij}$, $f_1 = \Phi_i$ and $f_2 = \Phi_{ij}^{old}$, since $S_{ij} = \mathbf{C}_i \cap \mathbf{C}_j \subseteq \mathbf{C}_i$. Both $\Phi_{ij}^{new} = \sum_{\mathbf{C}_i\setminus S_{ij}} \Phi_i$ and Φ_{ij}^{old} are factors over S_{ij}, so their quotient is the ordinary division of Definition 7.7 and

$$M_{ij} = \left(\sum_{\mathbf{C}_i\setminus S_{ij}} \Phi_i\right) \Big/ \Phi_{ij}^{old} = \sum_{\mathbf{C}_i\setminus S_{ij}} \Phi_i/\Phi_{ij}^{old},$$

the right-hand quotient being the generalized division of this exercise. Writing Φ_{ij} for the separator factor before the update, the Hugin step at node i becomes

$$M_{ij} = \sum_{\mathbf{C}_i\setminus S_{ij}} \frac{\Phi_i}{\Phi_{ij}},$$
$$\Phi_{ij} \leftarrow \Phi_{ij} M_{ij},$$
$$\Phi_j \leftarrow \Phi_j M_{ij}.$$

that is, *divide first and marginalize afterwards* in place of the book's *marginalize first and divide afterwards*, with the separator now updated by multiplying the message in exactly as the receiving cluster is, so that the auxiliary copy Φ_{ij}^{old} disappears.
It is the same algorithm, by the following invariant of the two-pass schedule. Put $G_{ij} = \Psi_i \prod_{k\neq j} M_{ki}$. Then Φ_{ij} is the product of the messages j has so far sent to i, and

$$\Phi_i = \Phi_{ij}\, G_{ij}.$$

Indeed Φ_i changes only by $\Phi_i \leftarrow \Phi_i M_{ki}$ and Φ_{ij} only when a message crosses $i - j$; under the inward/outward schedule exactly one message crosses each edge in each direction and i has not yet sent to j, so the only change so far is from j's message to i, if sent. On the inward pass j has not sent, so $\Phi_{ij} = 1$ while i has heard from all $k \neq j$, giving $\Phi_i = G_{ij}$; on the outward pass j has sent, its message computed against $\Phi_{ij} = 1$, so $\Phi_{ij} = M_{ji}$ and $\Phi_i = \Psi_i \prod_k M_{ki} = M_{ji} G_{ij}$.
Given the invariant the separator update is exact: at $\mathbf{s}$ with $\Phi_{ij}(\mathbf{s}) \neq 0$ the division and multiplication cancel, and at $\Phi_{ij}(\mathbf{s}) = 0$ the left side is 0 while the right is $\sum_{\mathbf{c}_i\sim\mathbf{s}} \Phi_{ij}(\mathbf{s}) G_{ij}(\mathbf{c}_i) = 0$. So $\Phi_{ij} M_{ij} = \sum_{\mathbf{C}_i\setminus S_{ij}} \Phi_i$ identically, the reformulated step leaves the jointree exactly where the book's step would, and the guarantees $\Pr(\mathbf{C}_i, e) = \Phi_i$ and $\Pr(S_{ij}, e) = \Phi_{ij}$ at the end of the outward pass are untouched.

7 Inference by Conditioning

Contents

7.1 Exercises 8.1–8.7 98
7.2 Exercises 8.8–8.14 105

7.1 Exercises 8.1–8.7

Problem 8.1. Consider the Bayesian network and corresponding dtree given in Figure 8.13.
The Bayesian network has seven variables A, B, C, D, E, F, G with edges

$$A \to D, \quad B \to D, \quad B \to E, \quad C \to E,$$
$$C \to F, \quad D \to F, \quad E \to G, \quad F \to G.$$

Hence A, B, and C are roots and the seven families (CPTs) are $\{A\}$, $\{B\}$, $\{C\}$, $\{A, B, D\}$, $\{B, C, E\}$, $\{C, D, F\}$, $\{E, F, G\}$.
The dtree has thirteen nodes, numbered 1 through 13, with node 1 the root:

- node 1 has children 2 and 3
- node 2 has children 4 and 5
- node 3 is a leaf carrying the CPT of C (family C)
- node 4 has children 6 and 7
- node 5 is a leaf carrying the CPT of G (family EFG)
- node 6 is a leaf carrying the CPT of F (family CDF)
- node 7 has children 8 and 9
- node 8 is a leaf carrying the CPT of E (family BCE)
- node 9 has children 10 and 11
- node 10 is a leaf carrying the CPT of B (family B)
- node 11 has children 12 and 13
- node 12 is a leaf carrying the CPT of D (family ABD)
- node 13 is a leaf carrying the CPT of A (family A)

(a) Find vars(T), cutset(T), acutset(T), and context(T) for each node T in the dtree.
(b) What is the dtree width?

Solution. Part (a) is the table below and the width in part (b) is 3.
Compute vars bottom-up from the leaf CPTs by $\text{vars}(T) = \text{vars}(T^l) \cup \text{vars}(T^r)$:

$$\begin{aligned}
\text{vars}(13) &= \{A\}, & \text{vars}(12) &= \{A, B, D\},\\
\text{vars}(11) &= \{A, B, D\}, & \text{vars}(10) &= \{B\},\\
\text{vars}(9) &= \{A, B, D\}, & \text{vars}(8) &= \{B, C, E\},\\
\text{vars}(7) &= \{A, B, C, D, E\}, & \text{vars}(6) &= \{C, D, F\},\\
\text{vars}(5) &= \{E, F, G\}, & \text{vars}(4) &= \{A, B, C, D, E, F\},\\
\text{vars}(3) &= \{C\}, & \text{vars}(2) &= \{A, \ldots, G\},\\
\text{vars}(1) &= \{A, \ldots, G\}.
\end{aligned}$$

Then cutsets top-down by Definition 8.3, $\text{cutset}(T) = (\text{vars}(T^l) \cap \text{vars}(T^r)) \setminus \text{acutset}(T)$, each node's a-cutset being the union of its ancestors' cutsets and empty at the root:

$$\begin{aligned}
\text{cutset}(1) &= ABCDEFG \cap C = \{C\},\\
\text{cutset}(2) &= (ABCDEF \cap EFG) \setminus \{C\} = \{E, F\},\\
\text{cutset}(4) &= (CDF \cap ABCDE) \setminus \{C, E, F\} = \{D\},\\
\text{cutset}(7) &= (BCE \cap ABD) \setminus \{C, D, E, F\} = \{B\},\\
\text{cutset}(9) &= (B \cap ABD) \setminus \{B, C, D, E, F\} = \emptyset,\\
\text{cutset}(11) &= (ABD \cap A) \setminus \{B, C, D, E, F\} = \{A\}.
\end{aligned}$$

Contexts are $\text{vars}(T) \cap \text{acutset}(T)$ (Definition 8.5), and clusters are vars(T) at a leaf and $\text{cutset}(T) \cup \text{context}(T)$ otherwise (Definition 8.6):

T	vars(T)	cutset(T)	acutset(T)	context(T)	cluster(T)
1	$ABCDEFG$	C	$\emptyset$	$\emptyset$	C
2	$ABCDEFG$	EF	C	C	CEF
3	C	–	C	C	C
4	$ABCDEF$	D	CEF	CEF	$CDEF$
5	EFG	–	CEF	EF	EFG
6	CDF	–	$CDEF$	CDF	CDF
7	$ABCDE$	B	$CDEF$	CDE	$BCDE$
8	BCE	–	$BCDEF$	BCE	BCE
9	ABD	$\emptyset$	$BCDEF$	BD	BD
10	B	–	$BCDEF$	B	B
11	ABD	A	$BCDEF$	BD	ABD
12	ABD	–	$ABCDEF$	ABD	ABD
13	A	–	$ABCDEF$	A	A

A dash means the node is a leaf and so carries no cutset.

Part (b). The largest clusters are cluster(4) = $CDEF$ and cluster(7) = $BCDE$, of size 4, so the width is $4-1=3$. This is optimal: eliminating A, G, B, C, D, E, F from the moral graph gives clusters ABD, EFG, $BCDE$, $CDEF$, DEF, EF, F, so the treewidth is 3.

Problem 8.2. Consider the dtree in Figure 8.13 (the network has variables $A, \ldots, G$ with edges $A \to D$, $B \to D$, $B \to E$, $C \to E$, $C \to F$, $D \to F$, $E \to G$, $F \to G$; the thirteen-node dtree has root 1 with children 2, 3; node 2 with children 4, 5; node 4 with children 6, 7; node 7 with children 8, 9; node 9 with children 10, 11; node 11 with children 12, 13; and leaves 3 = C, 5 = EFG, 6 = CDF, 8 = BCE, 10 = B, 12 = ABD, 13 = A). Assuming that all variables are binary:
(a) How many recursive calls will RC1 make when run on this dtree with no evidence?
(b) How many recursive calls will RC2 make when run on this dtree with no evidence?
(c) How many cache hits will occur at node 9 when RC2 is run with no evidence?
(d) Which of the dtree nodes have dead caches, if any?
All of these questions can be answered without tracing the recursive conditioning algorithm.

Solution. (a) 309 calls, (b) 109, (c) 12 hits, (d) the nonleaf nodes 2, 4, 11 (and the root). Throughout, the cutsets, a-cutsets, contexts and clusters are those of Exercise 8.1, and $\mathbf{X}^{\#} = 2^{|\mathbf{X}|}$ counts instantiations.

Part (a). Lines 6–7 of RC1 loop over the instantiations of cutset(T) compatible with the evidence, one call to each child per instantiation; with empty evidence none is excluded, so

$$\text{calls}(T^l) = \text{calls}(T^r) = \text{calls}(T) \cdot \text{cutset}(T)^{\#},$$

with calls(root) = 1, whose solution is calls(T) = acutset(T)$^{\#}$ — the bound of Theorem 8.1, met with equality since the evidence is empty:

node	1	2	3	4	5	6	7	8	9	10	11	12	13
a-cutset size	0	1	1	3	3	4	4	5	5	5	5	6	6
calls	1	2	2	8	8	16	16	32	32	32	32	64	64

Summing,

$$1 + 2 \cdot 2 + 2 \cdot 8 + 2 \cdot 16 + 4 \cdot 32 + 2 \cdot 64 = 1 + 4 + 16 + 32 + 128 + 128 = 309.$$

that is, 308 recursive calls beyond the top-level call on the root.

Part (b). RC2 caches at every nonleaf node, so T executes its loop once per instantiation of context(T) and answers all later calls from the cache; with empty evidence every such instantiation is generated, so the loop runs $\min(\text{calls}(T), \text{context}(T)^{\#})$ times and each child of T receives

$$\text{cutset}(T)^{\#}\,\text{context}(T)^{\#} = \text{cluster}(T)^{\#}$$

calls, the bound of Theorem 8.3 attained everywhere. Reading the parents' clusters off Exercise 8.1:

node	1	2	3	4	5	6	7	8	9	10	11	12	13
calls	1	2	2	8	8	16	16	16	16	4	4	8	8

summing to

$$1 + 2(2 + 8 + 16 + 16 + 4 + 8) = 1 + 2 \cdot 54 = 109,$$

against 309 for RC1.

Part (c). Node 9 receives 16 calls, and context$(9) = \{B, D\}$ has 4 instantiations, all occurring under empty evidence. The first call carrying a given instantiation misses and fills that entry, so there are 4 misses and

$$16 - 4 = 12$$

cache hits at node 9.

Part (d). By Equation 8.4 the cache at T is dead when context(T) = cluster(T^p), since then the cluster$(T^p)^{\#}$ calls T receives all carry distinct instantiations of context(T) and no entry is looked up after being filled. Checking each nonroot node against its parent's cluster:

T	context(T)	cluster(T^p)	dead?
2	C	C	yes
3	C	C	yes (leaf)
4	CEF	CEF	yes
5	EF	CEF	no
6	CDF	$CDEF$	no
7	CDE	$CDEF$	no
8	BCE	$BCDE$	no
9	BD	$BCDE$	no
10	B	BD	no
11	BD	BD	yes
12	ABD	ABD	yes (leaf)
13	A	ABD	no

RC2 does not cache at leaves (they are answered by LOOKUP, a tabular CPT serving as its own cache), so the dead caches are those of the nonleaf nodes 2, 4, 11, together with the root, which is called exactly once and so never retrieves its single entry.

Problem 8.3. Figure 8.13 depicts a Bayesian network and a corresponding dtree with height 6. The network has variables $A, \ldots, G$ with edges

$$A \to D, \quad B \to D, \quad B \to E, \quad C \to E,$$
$$C \to F, \quad D \to F, \quad E \to G, \quad F \to G,$$

so its seven CPTs have families $\{A\}$, $\{B\}$, $\{C\}$, $\{A, B, D\}$, $\{B, C, E\}$, $\{C, D, F\}$, $\{E, F, G\}$. The dtree of Figure 8.13 is the right-deep tree with root 1 (children $2, 3$), node 2 (children $4, 5$), node 4 (children $6, 7$), node 7 (children $8, 9$), node 9 (children $10, 11$), node 11 (children $12, 13$), and leaves $3 = C$, $5 = EFG$, $6 = CDF$, $8 = BCE$, $10 = B$, $12 = ABD$, $13 = A$; its width is 3 (Exercise 8.1). Find a dtree for this network whose height is 3 and whose width is no worse than the width of the original dtree.

Solution. Take the dtree with root R and internal nodes L, M, L_1, L_2, M_1:

- R has children L and M
- L has children L_1 and L_2
- L_1 has children the leaves A and ABD
- L_2 has children the leaves B and BCE
- M has children the leaf EFG and the node M_1
- M_1 has children the leaves C and CDF

Every node has 0 or 2 children and the seven leaves are exactly the seven CPTs, so this is a dtree (Definition 8.2); the leaf EFG sits at depth 2 and the other six at depth 3, so the height is 3, and the

computation below gives width 3.
The variable sets are

$$\begin{aligned} \text{vars}(L_1) &= \{A, B, D\}, & \text{vars}(L_2) &= \{B, C, E\}, \\ \text{vars}(L) &= \{A, B, C, D, E\}, & \text{vars}(M_1) &= \{C, D, F\}, \\ \text{vars}(M) &= \{C, D, E, F, G\}, & \text{vars}(R) &= \{A, \ldots, G\}. \end{aligned}$$

At the root, $\text{acutset}(R) = \emptyset$ and

$$\text{cutset}(R) = \text{vars}(L) \cap \text{vars}(M) = \{A, B, C, D, E\} \cap \{C, D, E, F, G\} = \{C, D, E\},$$

so $\text{cluster}(R) = CDE$ and $\text{acutset}(L) = \text{acutset}(M) = \{C, D, E\}$. Then

$$\begin{aligned} \text{cutset}(L) &= \big(\{A, B, D\} \cap \{B, C, E\}\big) \setminus \{C, D, E\} = \{B\}, \\ \text{context}(L) &= \{A, B, C, D, E\} \cap \{C, D, E\} = \{C, D, E\}, \end{aligned}$$

giving $\text{cluster}(L) = \{B, C, D, E\}$, and

$$\begin{aligned} \text{cutset}(M) &= \big(\{E, F, G\} \cap \{C, D, F\}\big) \setminus \{C, D, E\} = \{F\}, \\ \text{context}(M) &= \{C, D, E, F, G\} \cap \{C, D, E\} = \{C, D, E\}, \end{aligned}$$

giving $\text{cluster}(M) = \{C, D, E, F\}$. Descending once more, $\text{acutset}(L_1) = \text{acutset}(L_2) = \{B, C, D, E\}$ and $\text{acutset}(M_1) = \{C, D, E, F\}$, so

$$\begin{aligned} \text{cutset}(L_1) &= \big(\{A\} \cap \{A, B, D\}\big) \setminus \{B, C, D, E\} = \{A\}, \\ \text{context}(L_1) &= \{A, B, D\} \cap \{B, C, D, E\} = \{B, D\}, \\ \text{cutset}(L_2) &= \big(\{B\} \cap \{B, C, E\}\big) \setminus \{B, C, D, E\} = \emptyset, \\ \text{context}(L_2) &= \{B, C, E\} \cap \{B, C, D, E\} = \{B, C, E\}, \\ \text{cutset}(M_1) &= \big(\{C\} \cap \{C, D, F\}\big) \setminus \{C, D, E, F\} = \emptyset, \\ \text{context}(M_1) &= \{C, D, F\} \cap \{C, D, E, F\} = \{C, D, F\}. \end{aligned}$$

In full:

T	cutset(T)	acutset(T)	context(T)	cluster(T)	size
R	CDE	$\emptyset$	$\emptyset$	CDE	3
L	B	CDE	CDE	$BCDE$	4
M	F	CDE	CDE	$CDEF$	4
L_1	A	$BCDE$	BD	ABD	3
L_2	$\emptyset$	$BCDE$	BCE	BCE	3
M_1	$\emptyset$	$CDEF$	CDF	CDF	3
leaf EFG	–	$CDEF$	EF	EFG	3
leaf A	–	$ABCDE$	A	A	1
leaf ABD	–	$ABCDE$	ABD	ABD	3
leaf B	–	$BCDE$	B	B	1
leaf BCE	–	$BCDE$	BCE	BCE	3
leaf C	–	$CDEF$	C	C	1
leaf CDF	–	$CDEF$	CDF	CDF	3

The largest clusters are $\text{cluster}(L) = BCDE$ and $\text{cluster}(M) = CDEF$, of size 4, so by Definition 8.6 the width is $4 - 1 = 3$: no worse than the original, and optimal, the treewidth being 3 (Exercise 8.1).

Problem 8.4. Show a class of networks on which the time complexity of Algorithm RC1 is worse than the time complexity of cutset conditioning. Show also a class of networks on which the time complexity of cutset conditioning is worse than RC1.

Solution. RC1 is worse on polytrees; cutset conditioning is worse on chains of diamonds.
The two complexities: with n nodes, a loop-cutset of size s (Definition 8.1) and at most k parents per node, cutset conditioning makes $O(\exp(s))$ calls to the polytree algorithm at $O(n\exp(k))$ each, hence $O(n\exp(k+s))$ time; while RC1 on a balanced dtree of cutset width w takes $O(n\exp(w\log n)) = n^{O(w)}$ by Theorem 8.2. Neither dominates: either of s, w can be small while the other is large.
Class 1: RC1 worse. Take the chain

$$X_1 \to X_2 \to \cdots \to X_n,$$

a polytree, so $s = 0$, with $k = 1$; cutset conditioning is then a single run of the polytree algorithm, $\Theta(n)$ time.
RC1 costs $\Theta(n^2)$. The n CPTs are $\{X_1\}$ and $\{X_{i-1}, X_i\}$, and the balanced dtree of Figure 8.7 (right) splits each subchain at a midpoint variable, so every internal cutset is a single variable; a node at depth d then has $|\text{acutset}| = d$ and receives 2^d calls (Theorem 8.1), and there are 2^d such nodes. With depth $\log_2 n$,

$$\sum_{d=0}^{\log_2 n} 2^d \cdot 2^d = \sum_{d=0}^{\log_2 n} 4^d = \Theta(4^{\log_2 n}) = \Theta(n^2).$$

No dtree does better. For any dtree over the chain whose internal cutsets all have size 1, a leaf at depth d_i takes 2^{d_i} calls while the Kraft equality $\sum_i 2^{-d_i} = 1$ holds over the n leaves, so Cauchy–Schwarz gives

$$\Big(\sum_i 2^{d_i}\Big)\Big(\sum_i 2^{-d_i}\Big) \;\geq\; \Big(\sum_i 1\Big)^2 = n^2,$$

so at least n^2 calls go to leaves alone; and unbalancing buys an occasional empty cutset only by deepening other leaves — in the extreme the right-deep dtree of Figure 8.7 (left) has a-cutset width $n-1$ and $\Theta(2^n)$ calls. So on chains RC1 is asymptotically the worse algorithm, because it caches nothing and re-solves each subnetwork once per instantiation of its a-cutset.
Class 2: cutset conditioning worse. Take the networks of Figure 8.4: a chain of n diamonds on middle variables $M_0, \ldots, M_n$, top variables $T_1, \ldots, T_n$ and bottom variables $B_1, \ldots, B_n$, with edges

$$M_{i-1} \to T_i, \quad T_i \to M_i, \quad M_{i-1} \to B_i, \quad B_i \to M_i \qquad (i = 1, \ldots, n),$$

so that consecutive diamonds share a middle variable. There are $3n+1$ nodes and $k = 2$, only the M_i with $i \geq 1$ having two parents.
The i-th diamond is the edge-disjoint undirected cycle M_{i-1}, T_i, M_i, B_i, and by Definition 8.1 a loop-cutset must break each by deleting outgoing edges. On cycle i the outgoing edges belong to M_{i-1}, T_i, B_i (the node M_i has none), so the cutset contains one of these three for every i; and each of them has all its outgoing edges on cycle i alone, so no node serves two cycles. Hence $s \geq n$, and with binary variables cutset conditioning considers at least 2^n cases: $\Omega(2^n)$ time.
RC1 is polynomial here. Moralizing adds only the edges $T_i - B_i$, the M_i being the sole nodes with two parents, and eliminating in the order $M_0, T_1, B_1, M_1, T_2, \ldots$ creates no fill-in: M_{i-1} leaves T_i, B_i, adjacent through the moral edge, cluster $\{M_{i-1}, T_i, B_i\}$; T_i leaves B_i, M_i, adjacent through $B_i \to M_i$, cluster $\{T_i, B_i, M_i\}$; B_i leaves only M_i. Every cluster has size 3, so the treewidth is 2 whatever n, and the construction of Chapter 9 yields a balanced dtree of cutset width $w \leq 3$, for which Theorem 8.2 bounds RC1 by

$$O\big(n\exp(w\log n)\big) = n^{O(1)}.$$

Problem 8.5. Show that for networks whose loop-cutset is bounded, Algorithm RC2 will have time and space complexity that are linear in the network size if RC2 is run on a carefully chosen dtree.

Solution. Take a dtree T' of small width for the polytree and transplant its shape, relabelling the leaves with the true families; the result has width at most $k+s$, and Theorem 8.4 — RC2 on a width-w dtree costs $O(n\exp(w))$ time and space — then gives the claim.

Write $F_X = \{X\} \cup \mathbf{U}_X$ for the family of X in $\mathcal{N}$, $\mathbf{C}$ for a loop-cutset (Definition 8.1) of size s, k for the maximum number of parents, and $\mathcal{N}_{\mathbf{C}}$ for the polytree obtained by deleting the edges out of $\mathbf{C}$, in which X has family $F'_X = \{X\} \cup (\mathbf{U}_X \setminus \mathbf{C})$, so that

$$F_X \setminus \mathbf{C} \;=\; F'_X \setminus \mathbf{C}, \qquad F'_X \subseteq F_X, \qquad F_X \subseteq F'_X \cup \mathbf{C}.$$

Step 1. $\mathcal{N}_{\mathbf{C}}$ has an elimination order of width $k' := \max_X |F'_X| - 1 \leq k$: moralize and repeatedly eliminate a leaf X of the underlying undirected tree, whose neighbours are either its parents $\mathbf{U}'_X$, made pairwise adjacent by moralization, or (if X is parentless with one child Y) the uneliminated part of $F'_Y \setminus \{X\}$, again a clique; either way no fill-in and a cluster inside a family. By the polytime construction of Chapter 9 there is then a dtree T' for $\mathcal{N}_{\mathbf{C}}$ of width $\leq k'$, its leaves labelled F'_X.
Step 2. Let T have the same shape as T' with the leaf of X relabelled by the true family F_X. This is the carefully chosen dtree: a full binary tree whose leaves correspond one-to-one with the CPTs of $\mathcal{N}$ (Definition 8.2).
Step 3. Fix a tree shape and two labellings of its leaves, by sets L_i and by $L_i \setminus \mathbf{C}$, writing vars^- and so on for the second. Since $\text{vars}(T)$ is the union of the leaf labels below T, $\text{vars}^-(T) = \text{vars}(T) \setminus \mathbf{C}$ at every node, and we claim

$$\text{cutset}^-(T) = \text{cutset}(T) \setminus \mathbf{C}, \qquad \text{acutset}^-(T) = \text{acutset}(T) \setminus \mathbf{C}.$$

by induction on depth: both a-cutsets are empty at the root, and given $\text{acutset}^-(T) = \text{acutset}(T) \setminus \mathbf{C}$, Definition 8.3 gives

$$\begin{aligned}
\text{cutset}^-(T) &= \big(\text{vars}^-(T^l) \cap \text{vars}^-(T^r)\big) \setminus \text{acutset}^-(T) \\
&= \Big(\big(\text{vars}(T^l) \cap \text{vars}(T^r)\big) \setminus \mathbf{C}\Big) \setminus \big(\text{acutset}(T) \setminus \mathbf{C}\big) \\
&= \big(\text{vars}(T^l) \cap \text{vars}(T^r)\big) \setminus \big(\mathbf{C} \cup \text{acutset}(T)\big) \\
&= \text{cutset}(T) \setminus \mathbf{C},
\end{aligned}$$

the third equality because removing $\mathbf{C}$ and then $\text{acutset}(T) \setminus \mathbf{C}$ removes exactly $\mathbf{C} \cup \text{acutset}(T)$; a child's a-cutset being its parent's a-cutset together with its cutset, the induction goes through. Consequently

$$\text{context}^-(T) = \text{vars}^-(T) \cap \text{acutset}^-(T) = \text{context}(T) \setminus \mathbf{C},$$

and therefore $\text{cluster}^-(T) = \text{cluster}(T) \setminus \mathbf{C}$ at every node by Definition 8.6, leaves included.
Reduce both T and T' by $\mathbf{C}$. Their shapes agree and their reduced leaf labels agree, $F_X \setminus \mathbf{C} = F'_X \setminus \mathbf{C}$, so the two reductions are the same labelled tree and

$$\text{cluster}_T(T) \setminus \mathbf{C} \;=\; \text{cluster}_{T'}(T) \setminus \mathbf{C} \;\subseteq\; \text{cluster}_{T'}(T),$$

and so

$$|\text{cluster}_T(T)| \leq |\text{cluster}_{T'}(T)| + |\mathbf{C}| \leq (k' + 1) + s.$$

The width of T is therefore at most $k' + s \leq k + s$.
Step 4. By Theorem 8.4, RC2 on T takes

$$O\big(n \exp(k + s)\big) = O\big(\exp(s) \cdot n \exp(k)\big)$$

in both time and space. With the loop-cutset bounded, $s = O(1)$, this is $O(n \exp(k))$ — the complexity of the polytree algorithm, and linear in the size of the network, since a network with n nodes and at most k parents each has $O(n \exp(k))$ CPT parameters.

Problem 8.6. Show that if a network is pruned as given in Section 6.9 before applying RC1, then Line 6 of algorithm LOOKUP will never be reached. Recall that pruning a network can only be done under a given query.

For reference, LOOKUP$(T, \mathbf{e})$ is

1. $\Theta_{X|\mathbf{U}} \leftarrow$ CPT associated with dtree node T
2. $\mathbf{u} \leftarrow$ instantiation of $\mathbf{U}$ compatible with evidence $\mathbf{e}$
3. if X is instantiated to x in evidence $\mathbf{e}$ then
4. return $\theta_{x|\mathbf{u}}$
5. else
6. return 1 (that is, $\sum_x \theta_{x|\mathbf{u}}$)
7. end if

Solution. Suppose Line 6 is reached at a leaf dtree node T carrying $\Theta_{X|\mathbf{U}}$; two facts force a contradiction. Since RC1 computes $\Pr(\mathbf{e})$, the query variables are $\mathbf{Q} = \emptyset$, and pruning under that query (Section 6.9) prunes the edges outgoing from the evidence nodes $\mathbf{E}$ and repeatedly deletes leaf nodes outside $\mathbf{Q} \cup \mathbf{E} = \mathbf{E}$, leaving a network $\mathcal{N}^\star$ for which the dtree is built.
Fact 1: every leaf node of $\mathcal{N}^\star$ is in $\mathbf{E}$, node pruning having run until no such node remains.
Fact 2: the evidence RC1 passes to LOOKUP at T is $\mathbf{ec}$, with $\mathbf{c}$ an instantiation of acutset(T), since Line 7 extends the evidence by the current cutset instantiation at each call and the cutsets along the root-to-T path are exactly acutset(T) (Definition 8.3). So Line 3 succeeds exactly when $X \in \mathbf{E} \cup$ acutset(T), and Line 6 is reached only if $X \notin \mathbf{E}$ and $X \notin$ acutset(T).
The latter forces X into a single dtree leaf. Let $\mathcal{S}$ be the leaves whose CPT mentions X and suppose $|\mathcal{S}| \geq 2$, with deepest common ancestor $T^\star$. Then $T^\star$ is a nonleaf with a member of $\mathcal{S}$ in each subtree, so $X \in \text{vars}(T^{\star l}) \cap \text{vars}(T^{\star r})$; and for any proper ancestor A of $T^\star$ all of $\mathcal{S}$ lies in one of A's subtrees, so X is missing from the other and $X \notin$ cutset(A), i.e. $X \notin$ acutset$(T^\star)$. Definition 8.3 then gives $X \in$ cutset$(T^\star) \subseteq$ acutset(T), against Fact 2. Hence $|\mathcal{S}| = 1$.
But X occurs in its own CPT and in that of each of its children in $\mathcal{N}^\star$, so $|\mathcal{S}| = 1$ makes X a leaf node of $\mathcal{N}^\star$ and thus $X \in \mathbf{E}$ by Fact 1, contradicting $X \notin \mathbf{E}$. So Line 6 is never reached.

Problem 8.7. Consider the Bayesian network and dtree shown in Figure 8.14. The network has four variables with edges

$$A \to C, \quad B \to C, \quad B \to D, \quad C \to D,$$

so its four families (CPTs) are $\{A\}$, $\{B\}$, $\{A, B, C\}$, $\{B, C, D\}$. The dtree has seven nodes:

- the root R has children N_2 and the leaf BCD
- N_2 has children N_3 and the leaf B
- N_3 has children the leaf A and the leaf ABC

Create a dgraph for this network by orienting the given dtree with respect to each of its leaf nodes, that is, by running Algorithm DT2DG.

Solution. Stage 1. Line 2 of DT2DG drops the edge directions, removes the dtree root and joins its neighbors. The dtree edges are

$$R - N_2, \quad R - BCD, \quad N_2 - N_3, \quad N_2 - B, \quad N_3 - A, \quad N_3 - ABC.$$

so deleting R and joining N_2 to BCD leaves *Tree* with edges

$$N_2 - BCD, \quad N_2 - N_3, \quad N_2 - B, \quad N_3 - A, \quad N_3 - ABC.$$

Writing $c_1 = N_2$ and $c_2 = N_3$, this tree has four leaves A, ABC, B, BCD and two internal nodes of degree three:

$$c_1 \text{ adjacent to } \{BCD,\ B,\ c_2\}, \qquad c_2 \text{ adjacent to } \{A,\ ABC,\ c_1\}.$$

Stage 2. For a leaf P with neighbor C, DT2DG creates a root with children P and ORIENT(C, P), where ORIENT(C, P) returns C if C is a leaf and otherwise a node with children ORIENT(N_1, C), ORIENT(N_2, C) for the two neighbors $N_1, N_2 \neq P$ of C; results are cached under (C, P).
Toward A: here $C = c_2$, with other neighbors ABC and c_1, and the neighbors of c_1 other than c_2 are the leaves BCD, B, so

$$R_A = (\ A,\ X_1\ ), \qquad X_1 = (\ ABC,\ X_2\ ), \qquad X_2 = (\ BCD,\ B\ ).$$

Toward ABC: again $C = c_2$, now with other neighbors A and c_1, and ORIENT(c_1, c_2) hits the cache, returning X_2:

$$R_{ABC} = (\, ABC,\ X_3\,), \qquad X_3 = (\, A,\ X_2\,).$$

Toward B: here $C = c_1$, with other neighbors BCD and c_2, whose remaining neighbors are the leaves A, ABC, so ORIENT$(c_2, c_1) = X_5$:

$$R_B = (\, B,\ X_4\,), \qquad X_4 = (\, BCD,\ X_5\,), \qquad X_5 = (\, A,\ ABC\,).$$

Toward BCD: here $C = c_1$, with other neighbors B and c_2, and ORIENT(c_2, c_1) hits the cache, returning X_5:

$$R_{BCD} = (\, BCD,\ X_6\,), \qquad X_6 = (\, B,\ X_5\,).$$

The dgraph therefore has four roots $R_A, R_{ABC}, R_B, R_{BCD}$, six internal nodes $X_1, \ldots, X_6$, and the four shared leaves A, ABC, B, BCD:

node	children
R_A	A, X_1
R_{ABC}	ABC, X_3
R_B	B, X_4
R_{BCD}	BCD, X_6
X_1	ABC, X_2
X_2	BCD, B
X_3	A, X_2
X_4	BCD, X_5
X_5	A, ABC
X_6	B, X_5

The sharing is visible: X_2 has parents X_1, X_3 and X_5 has parents X_4, X_6, while each leaf has three parents. The counts match Theorem 8.6 for $n = 7$: $(5n - 7)/2 = 14$ nodes and $4(n - 2) = 20$ edges. The root cutsets deliver the family marginals. Computing vars bottom-up,

$$\begin{aligned} \text{vars}(X_2) &= \{B, C, D\}, & \text{vars}(X_5) &= \{A, B, C\}, \\ \text{vars}(X_1) &= \{A, B, C, D\}, & \text{vars}(X_3) &= \{A, B, C, D\}, \\ \text{vars}(X_4) &= \{A, B, C, D\}, & \text{vars}(X_6) &= \{A, B, C\}, \end{aligned}$$

so, since each root has an empty a-cutset,

$$\begin{aligned} \text{cutset}(R_A) &= \{A\} \cap \{A, B, C, D\} = \{A\}, \\ \text{cutset}(R_{ABC}) &= \{A, B, C\} \cap \{A, B, C, D\} = \{A, B, C\}, \\ \text{cutset}(R_B) &= \{B\} \cap \{A, B, C, D\} = \{B\}, \\ \text{cutset}(R_{BCD}) &= \{B, C, D\} \cap \{A, B, C\} = \{B, C\}. \end{aligned}$$

Since A, B, C all have children, their roots carry the full families $\{A\}$, $\{B\}$, $\{A, B, C\}$, and running RC there yields $\Pr(A, \mathbf{e})$, $\Pr(B, \mathbf{e})$, $\Pr(A, B, C, \mathbf{e})$; D is a network leaf, so its root carries only $\mathbf{U}_D = \{B, C\}$ and RC yields $\Pr(B, C, \mathbf{e})$, whence $\Pr(B, C, D, \mathbf{e}) = \Pr(B, C, \mathbf{e})\, \theta_{d|bc}$.

7.2 Exercises 8.8–8.14

Problem 8.8. Suppose that Algorithm DT2DG is passed a dtree with n nodes and height $O(\log n)$. Will the generated dgraph also have height $O(\log n)$?

Recall the algorithm. Given a dtree T with at least five nodes, DT2DG first builds an undirected tree Tree by dropping all edge directions, deleting the dtree root, and connecting the root's two neighbors by an edge. It then produces one dtree of the dgraph for each leaf P of T: the root of that dtree is a new node whose children are P and ORIENT(C, P), where C is the unique neighbor of P in Tree; and ORIENT(C, P) returns C itself if C is a leaf, otherwise a new node whose children are

ORIENT(N_1, C) and ORIENT(N_2, C), where N_1, N_2 are the two neighbors of C in Tree other than P. The results of ORIENT are cached, so the dtrees of the dgraph share structure. The height of a dgraph is the maximum height of the dtrees it contains.

Solution. Yes: the dgraph's height is at most twice the input dtree's height h, so $O(\log n)$ is preserved.
Step 1: in the dtree T_P obtained by orienting Tree toward a leaf P, a node created by ORIENT$(C, \cdot)$ — say it represents C — sits at depth $d_{\text{Tree}}(P, C)$, the root R having depth 0. By induction on depth: the children of R are P at depth 1 and ORIENT(C, P) for C the unique neighbor of P, at depth $1 = d_{\text{Tree}}(P, C)$; and if a node representing C sits at depth $d = d_{\text{Tree}}(P, C)$, created by ORIENT$(C, P')$, then P' is the neighbor of C on the path from P (the recursion only moves away from P), so for each of the two other neighbors N_i that path passes through C and

$$d_{\text{Tree}}(P, N_i) = d_{\text{Tree}}(P, C) + 1 = d + 1.$$

Those two neighbors always exist: every nonleaf node of Tree has degree 3, a nonroot internal dtree node keeping its parent and two children, and each neighbor of the deleted root gaining the other in exchange for it.
Step 2: the leaves of T_P are the leaves of Tree other than P, returned unchanged by ORIENT, together with the copy of P at depth 1. So by Step 1 the height of T_P is $\max_L d_{\text{Tree}}(P, L)$ over those leaves, the eccentricity of P, since the farthest node from any node of a tree is a leaf — hence at most diam(Tree).
Step 3: diam(Tree) $\le 2h$. In the undirected version of the input dtree the path between u and v runs through their deepest common ancestor, so has length at most $\text{depth}(u) + \text{depth}(v) \le 2h$; and deleting the root r while joining its two children replaces each subpath $u' - r - v'$ by a single edge, leaving other distances unchanged, so distances only decrease.
Hence every dtree in the dgraph has height at most $2h$, and $h = O(\log n)$ gives a dgraph of height $O(\log n)$.

Problem 8.9. Let $T_1, T_2, \ldots, T_n$ be a descending path in a dtree (that is, T_{i+1} is a child of T_i for $i = 1, \ldots, n-1$) where

$$cf(T_i) = \begin{cases} 1, & \text{for } i = 1 \text{ and } i = n \\ 0, & \text{otherwise.} \end{cases}$$

Show that each entry of $cache_{T_n}$ will be retrieved no more than

$$\Big(\big(\text{context}(T_1) \cup \bigcup_{i=1}^{n-1} \text{cutset}(T_i)\big) \setminus \text{context}(T_n)\Big)^{\#} - 1$$

times after it has been filled by Algorithm RC. Show also that the definition of a dead dtree cache given by Equation 8.4, namely that the cache at node T is dead when $\text{context}(T) = \text{cluster}(T^p)$ and we cache at T^p, follows as a special case.
Here $\mathbf{X}^{\#}$ denotes the number of instantiations of the variables $\mathbf{X}$, so that $\emptyset^{\#} = 1$.

Solution. Take the evidence empty, the worst case, since evidence only removes instantiations from the loop on Line 11 of RC. Write

$$\mathbf{W} \;=\; \text{context}(T_1) \cup \bigcup_{i=1}^{n-1} \text{cutset}(T_i), \qquad \mathbf{X} \;=\; \mathbf{W} \setminus \text{context}(T_n),$$

so the claimed bound is $\mathbf{X}^{\#} - 1$.
Part 1: $\mathbf{W}$ is a disjoint union. The cutsets of distinct nonleaf nodes are pairwise disjoint by Lemma 8.1(d); and $\text{context}(T_1) \subseteq \text{acutset}(T_1) \subseteq \text{acutset}(T_i)$ (Definition 8.5, and T_1 an ancestor of each T_i)

while Definition 8.3 makes $\mathrm{cutset}(T_i)$ exclude $\mathrm{acutset}(T_i)$, so $\mathrm{cutset}(T_i) \cap \mathrm{context}(T_1) = \emptyset$. Hence

$$\mathbf{W}^{\#} \;=\; \mathrm{context}(T_1)^{\#} \prod_{i=1}^{n-1} \mathrm{cutset}(T_i)^{\#}.$$

Part 2: the calls to T_n are labelled injectively by instantiations of $\mathbf{W}$. Since $cf(T_1) = 1$, the first call to T_1 carrying an instantiation $\mathbf{y}$ of $\mathrm{context}(T_1)$ executes Lines 9-14 and stores the result while every later call carrying $\mathbf{y}$ returns on Line 7, so T_1 is expanded at most $\mathrm{context}(T_1)^{\#}$ times, each expansion calling T_2 once per instantiation of $\mathrm{cutset}(T_1)$. For $1 < i \leq n-1$, $cf(T_i) = 0$, so no call to T_i returns on Line 7 and every one is expanded, again calling T_{i+1} once per instantiation of $\mathrm{cutset}(T_i)$. So each call to T_n is identified by the record

$$(\mathbf{y}, \mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_{n-1}),$$

with $\mathbf{y}$ the instantiation of $\mathrm{context}(T_1)$ at the expansion of T_1 that started the chain and $\mathbf{c}_i$ the instantiation of $\mathrm{cutset}(T_i)$ chosen on Line 11. By Part 1 such a record is exactly an instantiation of $\mathbf{W}$, distinct calls receive distinct records, and each record is a subinstantiation of the evidence $\mathbf{ec}$ passed to T_n in that call.
Part 3: $\mathrm{context}(T_n) \subseteq \mathbf{W}$. The ancestors of T_n are $T_1, \ldots, T_{n-1}$ together with those of T_1, so

$$\mathrm{acutset}(T_n) = \mathrm{acutset}(T_1) \cup \bigcup_{i=1}^{n-1} \mathrm{cutset}(T_i).$$

and $\mathrm{vars}(T_n) \subseteq \mathrm{vars}(T_1)$, T_n being a descendant of T_1. Hence

$$\begin{aligned} \mathrm{context}(T_n) &= \mathrm{vars}(T_n) \cap \mathrm{acutset}(T_n) \\ &\subseteq \big(\mathrm{vars}(T_1) \cap \mathrm{acutset}(T_1)\big) \cup \bigcup_{i=1}^{n-1} \mathrm{cutset}(T_i) \\ &= \mathrm{context}(T_1) \cup \bigcup_{i=1}^{n-1} \mathrm{cutset}(T_i) \;=\; \mathbf{W}. \end{aligned}$$

Part 4: counting one T_n-type. Fix an instantiation $\mathbf{z}$ of $\mathrm{context}(T_n)$, that is, one entry of $cache_{T_n}$. By Part 2 the calls of type $\mathbf{z}$ carry pairwise distinct instantiations of $\mathbf{W}$, and by Part 3 each agrees with $\mathbf{z}$ on $\mathrm{context}(T_n) \subseteq \mathbf{W}$; the number of instantiations of $\mathbf{W}$ extending a fixed one of $\mathrm{context}(T_n)$ is

$$\big(\mathbf{W} \setminus \mathrm{context}(T_n)\big)^{\#} = \mathbf{X}^{\#},$$

so T_n receives at most $\mathbf{X}^{\#}$ calls of type $\mathbf{z}$. Since $cf(T_n) = 1$, the first fills $cache_{T_n}[\mathbf{z}]$ on Line 14 and each later one retrieves it on Line 7, so the retrievals after filling number at most

$$\mathbf{X}^{\#} - 1 = \Big(\big(\mathrm{context}(T_1) \cup \bigcup_{i=1}^{n-1} \mathrm{cutset}(T_i)\big) \setminus \mathrm{context}(T_n)\Big)^{\#} - 1,$$

independently of the entry chosen, so for every entry of $cache_{T_n}$.
Part 5. Take $n = 2$, i.e. $T_1 = T^p$ and $T_2 = T$ with $cf(T^p) = cf(T) = 1$, the situation of Equation 8.4. Then

$$\mathbf{W} = \mathrm{context}(T^p) \cup \mathrm{cutset}(T^p) = \mathrm{cluster}(T^p)$$

by Definition 8.6, so each entry of $cache_T$ is retrieved at most

$$\big(\mathrm{cluster}(T^p) \setminus \mathrm{context}(T)\big)^{\#} - 1$$

times. Under the condition of Equation 8.4, $\mathrm{context}(T) = \mathrm{cluster}(T^p)$, that difference is $\emptyset$, with $\emptyset^{\#} = 1$, and the bound is 0: the cache at T is dead.

Problem 8.10. Show that the definition of a dead cache given by Equation 8.4 is not sufficient for a cache at a dgraph node T. That is, show an example where this condition is satisfied for every parent of dgraph node T yet cache entries at T are retrieved by RC after the cache has been filled. Recall that Equation 8.4 declares the cache at a dtree node T dead when

$$\text{context}(T) = \text{cluster}(T^p)$$

and we also cache at the parent T^p; recall also that a dgraph is a set of dtrees that share structure, that a dgraph node may therefore have several parents, that RC is run on a dgraph by running it on each of its root dtrees in turn, and that the cache attached to a shared dgraph node is shared by all the dtrees passing through it.

Solution. The dgraph of Figure 8.9 is already a counterexample: its shared node M satisfies Equation 8.4 at both parents, yet each of its four cache entries is retrieved once. Its network is that of Figure 8.10, with binary A, B, C, D and edges $A \to B$, $A \to C$, $B \to C$, $C \to D$, so the four CPTs — the leaves of every dtree in the dgraph — have variables A, AB, ABC, CD; Figure 8.9 is what DT2DG returns on the dtree of Figure 8.10, whose root has children N (children A, AB) and M_0 (children ABC, CD).

Step 1. Deleting that root and joining N to M_0 leaves the undirected tree Tree of Figure 8.11, with edges

$$N - A, \quad N - AB, \quad N - M_0, \quad M_0 - ABC, \quad M_0 - CD.$$

Orienting toward the leaf A gives

$$R_A \to \{\, A,\ Q_A \,\}, \qquad Q_A \to \{\, AB,\ M \,\}, \qquad M \to \{\, ABC,\ CD \,\},$$

where $M = \text{ORIENT}(M_0, N)$, and orienting toward the leaf AB gives

$$R_{AB} \to \{\, AB,\ Q_{AB} \,\}, \qquad Q_{AB} \to \{\, A,\ M \,\},$$

where ORIENT is called with (M_0, N) a second time and DT2DG's cache returns the very same node M. So M has the two parents Q_A, Q_{AB} and one cache $cache_M$ shared by both dtrees.

Step 2. With $\text{vars}(M) = ABCD$, Definitions 8.3, 8.5 and 8.6 give, top down in the dtree oriented toward A,

$$\begin{aligned}
\text{cutset}(R_A) &= \text{vars}(A) \cap \text{vars}(Q_A) = A \cap ABCD = A,\\
\text{context}(R_A) &= \emptyset, \qquad \text{cluster}(R_A) = A,\\
\text{cutset}(Q_A) &= \big(\text{vars}(AB) \cap \text{vars}(M)\big) \setminus A = AB \setminus A = B,\\
\text{context}(Q_A) &= \text{vars}(Q_A) \cap \text{acutset}(Q_A) = ABCD \cap A = A,\\
\text{cluster}(Q_A) &= B \cup A = AB,\\
\text{cutset}(M) &= (ABC \cap CD) \setminus AB = C,\\
\text{context}(M) &= ABCD \cap AB = AB.
\end{aligned}$$

In the dtree oriented toward AB,

$$\begin{aligned}
\text{cutset}(R_{AB}) &= \text{vars}(AB) \cap \text{vars}(Q_{AB}) = AB,\\
\text{context}(R_{AB}) &= \emptyset, \qquad \text{cluster}(R_{AB}) = AB,\\
\text{cutset}(Q_{AB}) &= \big(\text{vars}(A) \cap \text{vars}(M)\big) \setminus AB = A \setminus AB = \emptyset,\\
\text{context}(Q_{AB}) &= ABCD \cap AB = AB,\\
\text{cluster}(Q_{AB}) &= \emptyset \cup AB = AB,\\
\text{context}(M) &= ABCD \cap (AB \cup \emptyset) = AB.
\end{aligned}$$

The context of M is AB in both dtrees, so $cache_M$ is indexed by instantiations of AB and has four entries, as the sharing requires.

Step 3: Equation 8.4 holds at both parents, since

$$\text{cluster}(Q_A) = AB = \text{context}(M), \qquad \text{cluster}(Q_{AB}) = AB = \text{context}(M).$$

so if the condition were sufficient for a dgraph node, $cache_M$ could be deleted without penalty.
Step 4: the entries are nevertheless retrieved. Assume full caching, $cf \equiv 1$, empty evidence, and run RC on the roots in turn, R_A first.
In $RC(R_A, \text{true})$, the root is expanded once and Line 11 ranges over the two instantiations $\mathbf{a}$ of $\text{cutset}(R_A) = A$, calling $RC(Q_A, \mathbf{a})$; these carry distinct instantiations of $\text{context}(Q_A) = A$, so both miss and expand, each looping over the two instantiations $\mathbf{b}$ of $\text{cutset}(Q_A) = B$ and calling $RC(M, \mathbf{ab})$. The four calls carry the four distinct instantiations of $\text{context}(M) = AB$, so all four miss and fill $cache_M$ on Line 14 — no retrieval, as Equation 8.4 and Exercise 8.9 on the path Q_A, M predict.
In $RC(R_{AB}, \text{true})$, the root loops over the four instantiations $\mathbf{ab}$ of $\text{cutset}(R_{AB}) = AB$, calling $RC(Q_{AB}, \mathbf{ab})$; these carry the four distinct instantiations of $\text{context}(Q_{AB}) = AB$, so all miss and expand, and since $\text{cutset}(Q_{AB}) = \emptyset$ each expansion runs Line 11 once, issuing $RC(M, \mathbf{ab})$. Those four calls carry instantiations already stored from the first run, so all four are hits on Line 7: every entry of $cache_M$ is retrieved once after being filled, and Equation 8.4 is not sufficient for a dgraph cache.

Problem 8.11. Assume that CPTs support an implementation of LOOKUP that takes time and space that are linear in the number of CPT variables (instead of exponential). Provide a class of networks that have an unbounded treewidth yet on which RC will take time and space that are linear in the number of network variables.

Solution. Take $\mathcal{N}_k$ below: a single dense family hung with a long chain, of treewidth exactly k yet solved by RC in $O(n)$ time and space. The assumption buys only that a leaf T with CPT $\Theta_{X|\mathbf{U}}$ costs $O(|\mathbf{U}| + 1)$ on Line 2 of RC rather than a table of size $\Theta(\exp(|\mathbf{U}|))$, so leaves need no caches and a family may be arbitrarily large without the network being exponential; it does not reduce the number of recursive calls, and that is what must stay linear.
For $k = 1, 2, 3, \ldots$ let $\mathcal{N}_k$ be the network over binary variables

$$U_1, \ldots, U_k,\ X,\ Y_1, \ldots, Y_m, \qquad m = k\,2^k,$$

with edges $U_i \to X$, $X \to Y_1$, and $Y_j \to Y_{j+1}$ for $j = 1, \ldots, m-1$, so that

$$n \;=\; k + 1 + m \;=\; k + 1 + k\,2^k \;=\; \Theta(k\,2^k).$$

Moralizing makes $\{X, U_1, \ldots, U_k\}$ a clique, forcing treewidth $\geq k$, while the clusters $\{X, U_1, \ldots, U_k\}$, $\{X, Y_1\}, \{Y_1, Y_2\}, \ldots$ form a jointree of width k; so the treewidth is exactly k and the class is unbounded.
The leaves of the dtree are the $k + m + 1$ CPTs, namely $\{U_i\}$, the family $\{X, U_1, \ldots, U_k\}$, and $\{Y_1, X\}, \{Y_2, Y_1\}, \ldots, \{Y_m, Y_{m-1}\}$. Build

- D: the comb over the family part. Let D_0 be the leaf $\{X, U_1, \ldots, U_k\}$ and let D_i have children D_{i-1} and the leaf $\{U_i\}$; put $D = D_k$.
- C: any balanced dtree over the m chain leaves, as in the right-hand dtree of Figure 8.7.
- the root R with children D and C.

Since $\text{vars}(D) = \{X, U_1, \ldots, U_k\}$ and $\text{vars}(C) = \{X, Y_1, \ldots, Y_m\}$, Definition 8.3 gives $\text{cutset}(R) = \{X\}$: conditioning on X alone separates the dense family from the chain. Inside D, top down,

$$\text{cutset}(D_i) = \{U_i\}, \qquad \text{context}(D_i) = \{X, U_{i+1}, \ldots, U_k\},$$

while inside C every cutset is a single chain variable and every cluster has at most three variables, the balanced chain dtree having width 2 once X is counted.
Run RC with full caching at the internal nodes and none at the leaves. Every internal node being cached, Equation 8.5 collapses for a nonroot T to $calls(T) = \text{cutset}(T^p)^{\#}\,\text{context}(T^p)^{\#}$, the bound of Theorem 8.3 attained with equality, giving $2 \cdot 2^{k-i+1}$ calls to D_{i-1} and so

$$\begin{aligned} \text{calls inside } D &= \textstyle\sum_{i=0}^{k} \Theta(2^{k-i+1}) = \Theta(2^k), \\ \text{calls inside } C &= O(m \exp(2)) = O(m), \end{aligned}$$

the second by Theorem 8.4 on the width-2 balanced chain dtree, whose subproblems are solved once per value of X. The big leaf D_0 receives 2^{k+1} calls at $O(k)$ each by the assumed LOOKUP, contributing $O(k2^k) = O(n)$, so

$$\text{time} = O(2^k\, k + m) = O(k2^k) = O(n).$$

For space, the caches hold $\sum_T \text{context}(T)^{\#} = \Theta(2^k) + O(m) = O(n)$ entries, the dtree has $O(n)$ nodes, and by the assumption the CPTs occupy $O(k) + O(m) = O(n)$ rather than the $\Theta(2^k)$ a tabular $\Theta_{X|U_1 \ldots U_k}$ would need. Hence

$$\text{space} = O(n).$$

Problem 8.12. Suppose that you are running recursive conditioning under a limited amount of space. You have already run the depth-first branch-and-bound algorithm OPTIMAL CF and found the optimal discrete cache factor for your dgraph under a bound of M_{old} cache entries. Suddenly, the amount of space available to you increases to $M_{\text{new}} > M_{\text{old}}$. You want to find the new optimal discrete cache factor as quickly as possible.
(a) What is wrong with the following idea? Simply run the branch-and-bound algorithm on the set of nodes that are not included in the current cache factor, finding the subset of these nodes that will most efficiently fill up the new space. Then add these nodes to the current cache factor.
(b) How can we use our current cache factor to find the new optimal cache factor more quickly than if we had to start from scratch?

Solution. (a) The idea is unsound: it returns only cache factors extending cf_{old}, and the optimum for M_{new} need not extend the optimum for M_{old}.
(i) Wrong search space. Let $\text{context}(T)^{\#} = 8$ and $\text{context}(T')^{\#} = 4$, with caching at T alone removing 100 calls and at T' alone removing 20. Under $M_{\text{old}} = 4$ only T' fits, so cf_{old} caches at T'; under $M_{\text{new}} = 8$ the optimum caches at T alone, but the procedure is locked into T', has 4 entries left, cannot afford T, and misses the optimum by a factor of five. (Knapsack behaviour: an optimal small packing need not sit inside an optimal larger one.)
(ii) Ill-posed subproblem. Call reduction is not additive over nodes: by Equation 8.7 the gain from caching at T is

$$c_1 - c_2 = \text{cutset}(T)^{\#}\big(calls_{cf}(T) - \text{context}(T)^{\#}\big)\big(cpc_{cf}(T^l) + cpc_{cf}(T^r)\big),$$

which depends on the whole factor cf through $calls_{cf}$ and cpc_{cf} (Equation 8.5). Indeed caching at an ancestor T lowers $calls(T')$ at a descendant and can render $cache_{T'}$ dead (Equation 8.4 and Exercise 8.9), so memory already committed becomes waste. A correct algorithm must be free to withdraw a cache; this one is not.
(b) Reuse cf_{old} as an incumbent, not as a fixed core. Since $M_{\text{new}} > M_{\text{old}}$, every old-feasible factor is new-feasible, so $c^*_{\text{new}} \le c^*_{\text{old}}$, and we may rerun OPTIMAL CF with bound M_{new} after replacing Lines 1–2 by

$$cf^o \leftarrow cf_{\text{old}}, \qquad c^o \leftarrow c^*_{\text{old}}$$

in place of $c^o \leftarrow \infty$. Every search node with lower bound $\ge c^*_{\text{old}}$ is then pruned immediately on Line 1, where a fresh search must first descend to a complete factor; optimality survives because we prune only completions that cannot beat a factor in hand, and factors not extending cf_{old} are still explored. Three further savings. (i) If c^*_{old} equals lower bound($\emptyset$), the call count under full caching, then cf_{old} is optimal for any memory and no search is needed; in general stop as soon as an incumbent attains that value. (ii) Order Line 7 to prefer nodes cached by cf_{old}, ties by largest $\text{context}(T)^{\#}$ (Section 8.6.1), so the first dive spends the leftover memory and tightens the incumbent early. (iii) The dgraph, the lower-bound function and the quantities $calls(\cdot)$, $cpc(\cdot)$ of Section 8.6.2 do not depend on M, so the incremental update scheme starts from the old values.
Not reusable: the record of subtrees pruned for violating M_{old}, since such a subtree may hold the new optimum. Prunings from the bound test remain valid and are subsumed by the initialization above.

Problem 8.13. Consider the dtree in Figure 8.13. Assuming all variables are binary, which internal node will the greedy cache allocation algorithm GREEDY CF select first for caching? Show your work.

The Bayesian network of Figure 8.13 has binary variables A, B, C, D, E, F, G and edges

$$A \to D, \quad B \to D, \quad B \to E, \quad C \to E,$$
$$C \to F, \quad D \to F, \quad E \to G, \quad F \to G,$$

so its seven CPTs (the leaves of the dtree) have variable sets A, B, C, ABD, BCE, CDF, and EFG.

The accompanying dtree has thirteen nodes, numbered 1 through 13, with the following structure:

- node 1 (the root) has children 2 and 3; node 3 is the leaf C;
- node 2 has children 4 and 5; node 5 is the leaf EFG;
- node 4 has children 6 and 7; node 6 is the leaf CDF;
- node 7 has children 8 and 9; node 8 is the leaf BCE;
- node 9 has children 10 and 11; node 10 is the leaf B;
- node 11 has children 12 and 13; node 12 is the leaf ABD and node 13 is the leaf A.

Recall that GREEDY CF starts from the cache factor that caches nowhere and picks the node T maximizing $(c_1 - c_2)/\text{context}(T)^{\#}$, where c_1 and c_2 are the total numbers of recursive calls before and after caching at T; recall also that an internal node is one that is neither a leaf nor the root.

Solution. GREEDY CF selects node 9, with score 42, allocating it $\text{context}(9)^{\#} = 4$ cache entries.

Cutsets, contexts and clusters top down by Definitions 8.3, 8.5 and 8.6, using $\text{vars}(2) = ABCDEFG$, $\text{vars}(4) = ABCDEF$, $\text{vars}(7) = ABCDE$, $\text{vars}(9) = \text{vars}(11) = ABD$:

T	cutset(T)	acutset(T)	context(T)	cluster(T)	cutset(T)$^{\#}$	context(T)$^{\#}$
1	C	$\emptyset$	$\emptyset$	C	2	1
2	EF	C	C	CEF	4	2
3	leaf	C	C	C	-	2
4	D	CEF	CEF	$CDEF$	2	8
5	leaf	CEF	EF	EFG	-	4
6	leaf	$CDEF$	CDF	CDF	-	8
7	B	$CDEF$	CDE	$BCDE$	2	8
8	leaf	$BCDEF$	BCE	BCE	-	8
9	$\emptyset$	$BCDEF$	BD	BD	1	4
10	leaf	$BCDEF$	B	B	-	2
11	A	$BCDEF$	BD	ABD	2	4
12	leaf	$ABCDEF$	ABD	ABD	-	8
13	leaf	$ABCDEF$	A	A	-	2

(For instance $\text{cutset}(9) = (\text{vars}(10) \cap \text{vars}(11)) \setminus \text{acutset}(9) = B \setminus BCDEF = \emptyset$ and $\text{context}(9) = ABD \cap BCDEF = BD$; the rest is routine. Check!)

GREEDY CF starts from $cf_1 \equiv 0$, so Equation 8.5 collapses to $calls(T) = \text{cutset}(T^p)^{\#}\, calls(T^p)$ and Equation 8.6 to $cpc(T) = 1 + \text{cutset}(T)^{\#}(cpc(T^l) + cpc(T^r))$, with $cpc = 1$ at leaves. Traversing downward for *calls* and upward for *cpc*:

T	1	2	3	4	5	6	7	8	9	10	11	12	13
$calls(T)$	1	2	2	8	8	16	16	32	32	32	32	64	64
$cpc(T)$	309	153	1	37	1	1	17	1	7	1	5	1	1

Consistently, $c_1 = \sum_T calls(T) = 309 = cpc(1)$, the calls RC1 makes with no evidence.

The internal nodes are $2, 4, 7, 9, 11$, and Equation 8.7 gives

$$score_{cf_1}(T) = \frac{\text{cutset}(T)^{\#}\big(calls(T) - \text{context}(T)^{\#}\big)\big(cpc(T^l) + cpc(T^r)\big)}{\text{context}(T)^{\#}}.$$

Substituting the tables above:

$$\begin{aligned}
score(2) &= \frac{4\,(2-2)(37+1)}{2} = 0,\\
score(4) &= \frac{2\,(8-8)(1+17)}{8} = 0,\\
score(7) &= \frac{2\,(16-8)(1+7)}{8} = 16,\\
score(9) &= \frac{1\,(32-4)(1+5)}{4} = \frac{168}{4} = 42,\\
score(11) &= \frac{2\,(32-4)(1+1)}{4} = \frac{112}{4} = 28.
\end{aligned}$$

Directly: caching at T replaces $calls(T)$ by $\text{context}(T)^{\#}$ where T feeds its children, leaving counts above T unchanged, so recomputing $c_2 = \sum_S calls(S)$ gives

cached node T	c_1	c_2	$c_1 - c_2$	$M = \text{context}(T)^{\#}$	$(c_1 - c_2)/M$
2	309	309	0	2	0
4	309	309	0	8	0
7	309	181	128	8	16
9	309	141	168	4	42
11	309	197	112	4	28

(At node 9, caching turns its 32 calls into 4 expansions, so $10, 11$ drop from 32 to 4 and $12, 13$ from 64 to 8, giving $c_2 = 117 + 4 + 4 + 8 + 8 = 141$.) The direct counts match Equation 8.7 throughout, and the largest score is 42 at node 9.

The zeros are dead caches in the sense of Equation 8.4, since $\text{context}(2) = C = \text{cluster}(1)$ and $\text{context}(4) = CEF = \text{cluster}(2)$, so by Exercise 8.9 neither cache is ever read; deleting such nodes first leaves $7, 9, 11$ and again selects 9.

Problem 8.14. Prove Equation 8.7. That is, let cf_1 be a cache factor for a dgraph with $cf_1(T) = 0$ at some nonleaf node T, let cf_2 be the cache factor that results from caching at T (so $cf_2(T) = 1$ and $cf_2 = cf_1$ everywhere else), and let c_1 and c_2 be the total numbers of recursive calls made by RC under cf_1 and cf_2 as given by

$$calls(T) = \sum_{T^p} \text{cutset}(T^p)^{\#}\big[cf(T^p)\text{context}(T^p)^{\#} + (1 - cf(T^p))\, calls(T^p)\big],$$

which is Equation 8.5. With

$$cpc_{cf}(T) = \begin{cases} 1, & \text{if } T \text{ is a leaf or } cf(T) = 1\\ 1 + \text{cutset}(T)^{\#}\big(cpc_{cf}(T^l) + cpc_{cf}(T^r)\big), & \text{otherwise,}\end{cases}$$

which is Equation 8.6, show that

$$score_{cf_1}(T) = \frac{c_1 - c_2}{\text{context}(T)^{\#}} = \frac{\text{cutset}(T)^{\#}\big(calls_{cf_1}(T) - \text{context}(T)^{\#}\big)\big(cpc_{cf_1}(T^l) + cpc_{cf_1}(T^r)\big)}{\text{context}(T)^{\#}}.$$

Solution. Write V for the set of dgraph nodes and, for a cache factor cf, put

$$e_{cf}(S) \;=\; cf(S)\,\text{context}(S)^{\#} + (1 - cf(S))\, calls_{cf}(S),$$

the number of calls to S actually expanded (cf being discrete, this is $\text{context}(S)^{\#}$ when $cf(S) = 1$ and $calls_{cf}(S)$ when $cf(S) = 0$). Equation 8.5 then reads

$$calls_{cf}(S) \;=\; \sum_{S^p} \text{cutset}(S^p)^{\#}\, e_{cf}(S^p), \qquad (*)$$

with $calls_{cf}(S) = 1$ for a root S, independently of cf. The total number of recursive calls is $c = \sum_{S \in V} calls_{cf}(S)$.
Set $\delta(S) = calls_{cf_2}(S) - calls_{cf_1}(S)$ and $\varepsilon(S) = e_{cf_2}(S) - e_{cf_1}(S)$. Inducting in topological order, $\delta = 0$ at every root, and if S is not a proper descendant of T then no parent of S is T or a descendant of it, so $\delta = 0$ and (as $cf_1 = cf_2$ off T) $\varepsilon = 0$ at every parent, whence $\delta(S) = 0$ by $(*)$. In particular

$$calls_{cf_2}(T) = calls_{cf_1}(T),$$

(a DAG node is not a proper descendant of itself), while $cf_1(T) = 0$, $cf_2(T) = 1$ give

$$\varepsilon(T) \;=\; \mathrm{context}(T)^{\#} - calls_{cf_1}(T),$$

and $\varepsilon(S) = (1 - cf_1(S))\,\delta(S)$ for $S \neq T$. Unrolling $(*)$ along the DAG (terminating by acyclicity) therefore gives, for every S,

$$\delta(S) \;=\; \varepsilon(T) \sum_{\pi\,:\,T \rightsquigarrow S} w(\pi),$$

the sum ranging over all directed paths $\pi : T = S_0 \to S_1 \to \cdots \to S_r = S$ with $r \geq 1$, where

$$w(\pi) \;=\; \mathrm{cutset}(S_0)^{\#} \prod_{j=1}^{r-1} \big(1 - cf_1(S_j)\big)\,\mathrm{cutset}(S_j)^{\#}.$$

each application of $(*)$ pushing the perturbation to the children scaled by $\mathrm{cutset}(\cdot)^{\#}$, while a node with $cf_1(S_j) = 1$ absorbs it entirely ($e(S_j) = \mathrm{context}(S_j)^{\#}$ is independent of $calls(S_j)$). Paths are the right bookkeeping because a node may be reached from T by several routes, and by $(*)$ calls from different parents add.
Let $W(S)$ be the sum over all directed paths out of S, the empty path included, of $\prod_{j=0}^{r-1}(1 - cf_1(S_j))\mathrm{cutset}(S_j)^{\#}$ (empty path weight 1). Splitting off the first edge,

$$W(S) \;=\; 1 + \big(1 - cf_1(S)\big)\mathrm{cutset}(S)^{\#}\big(W(S^l) + W(S^r)\big),$$

with $W(S) = 1$ at a leaf. As cf_1 is discrete, $1 - cf_1(S)$ is 0 when $cf_1(S) = 1$ and 1 otherwise, so this recurrence is Equation 8.6 verbatim and $W = cpc_{cf_1}$ everywhere. Summing the path expansion over all nodes and grouping paths out of T by whether the first edge goes to T^l or T^r,

$$\begin{aligned} c_2 - c_1 = \sum_{S \in V} \delta(S) &= \varepsilon(T) \sum_{\pi \text{ starts at } T,\ |\pi| \geq 1} w(\pi) \\ &= \varepsilon(T)\,\mathrm{cutset}(T)^{\#}\big(W(T^l) + W(T^r)\big) \\ &= \varepsilon(T)\,\mathrm{cutset}(T)^{\#}\big(cpc_{cf_1}(T^l) + cpc_{cf_1}(T^r)\big). \end{aligned}$$

Substituting $\varepsilon(T) = \mathrm{context}(T)^{\#} - calls_{cf_1}(T)$ and changing sign,

$$c_1 - c_2 = \mathrm{cutset}(T)^{\#}\big(calls_{cf_1}(T) - \mathrm{context}(T)^{\#}\big)\big(cpc_{cf_1}(T^l) + cpc_{cf_1}(T^r)\big).$$

Dividing by the memory $M = \mathrm{context}(T)^{\#}$ that the greedy method allocates to $cache_T$ yields Equation 8.7:

$$score_{cf_1}(T) = \frac{c_1 - c_2}{\mathrm{context}(T)^{\#}} = \frac{\mathrm{cutset}(T)^{\#}\big(calls_{cf_1}(T) - \mathrm{context}(T)^{\#}\big)\big(cpc_{cf_1}(T^l) + cpc_{cf_1}(T^r)\big)}{\mathrm{context}(T)^{\#}}. \qquad \blacksquare$$

8 Models for Graph Decomposition

Contents

8.1 Exercises 9.1–9.7 115
8.2 Exercises 9.8–9.14 122
8.3 Exercises 9.15–9.21 126
8.4 Exercises 9.22–9.28 132

8.1 Exercises 9.1–9.7

Problem 9.1. Construct a jointree for the DAG in Figure 9.28 using the elimination order $\pi = A, G, B, C, D, E, F$ and Algorithm 23.
The DAG of Figure 9.28 has nodes A, B, C, D, E, F, G and edges

$$A \to D, \quad B \to D, \quad B \to E, \quad C \to E,$$

$$C \to F, \quad D \to F, \quad E \to G, \quad F \to G.$$

Its families are therefore A, B, C, ABD, BCE, CDF, EFG.

Solution. Algorithm 23 returns the chain $ABD - BCDE - CDEF - EFG$, with separators BD, CDE, EF.
Moralizing marries the parents of D, E, F, G, adding $A - B$, $B - C$, $C - D$, $E - F$, so G^m has adjacency

node	neighbours in G^m
A	B, D
B	A, C, D, E
C	B, D, E, F
D	A, B, C, F
E	B, C, F, G
F	C, D, E, G
G	E, F

By Definition 9.4, C_i is $\pi(i)$ with its current neighbours, which are then made a clique. Eliminating A (neighbours B, D, already adjacent) and G (neighbours E, F, already adjacent) creates no fill-in; eliminating B (remaining neighbours C, D, E, with $D - E$ missing) adds the fill-in $D - E$, after which C, D, E, F are pairwise adjacent and the remaining eliminations add nothing. The induced sequence is

$$C_1 = ABD, \quad C_2 = EFG, \quad C_3 = BCDE, \quad C_4 = CDEF,$$
$$C_5 = DEF, \quad C_6 = EF, \quad C_7 = F.$$

so $\text{width}(\pi, G) = 3$ by Definition 9.5, and every family lies in a cluster as Theorem 9.5 guarantees ($A, B, ABD \subseteq C_1$; $C, BCE \subseteq C_3$; $CDF \subseteq C_4$; $EFG = C_2$).
Each of $C_5 = DEF$, $C_6 = EF$, $C_7 = F$ is contained in $C_4 = CDEF$, which is in each case the largest index below with this property, so Theorem 9.7 deletes them one at a time, moving C_4 into the vacated adjacent position and thus leaving the order unchanged:

$$C_1 = ABD, \quad C_2 = EFG, \quad C_3 = BCDE, \quad C_4 = CDEF.$$

None of the four contains another, and the running intersection property of Theorem 9.6 survives:

$$C_1 \cap (C_2 \cup C_3 \cup C_4) = BD \subseteq C_3,$$
$$C_2 \cap (C_3 \cup C_4) = EF \subseteq C_4,$$
$$C_3 \cap C_4 = CDE \subseteq C_4.$$

Theorem 9.8 now attaches each C_i, working backwards from C_4, to a later cluster containing $C_i \cap (C_{i+1} \cup \cdots \cup C_4)$: (i) $CDE \subseteq C_4$, giving $BCDE - CDEF$; (ii) EF sits only in C_4 ($BCDE$ lacks F), giving $EFG - CDEF$; (iii) BD sits only in C_3, giving $ABD - BCDE$. The jointree is the chain

$$ABD \overset{BD}{—} BCDE \overset{CDE}{—} CDEF \overset{EF}{—} EFG,$$

separators written above their edges. Definition 9.13 holds: families lie in clusters as checked, and each variable spans a subpath of the chain (D in $ABD, BCDE, CDEF$; E in $BCDE, CDEF, EFG$; the rest in one or two adjacent clusters). Width $3 = \text{width}(\pi, G)$.

Problem 9.2. Construct a jointree for the DAG in Figure 9.28 using the elimination order $\pi = A, G, B, C, D, E, F$ and Algorithm 24. In case of multiple maximum spanning trees, show all of them.
The DAG of Figure 9.28 has nodes A, B, C, D, E, F, G and edges

$$A \to D, \quad B \to D, \quad B \to E, \quad C \to E,$$

$$C \to F, \quad D \to F, \quad E \to G, \quad F \to G.$$

Solution. The maximum spanning tree is unique, namely the chain $ABD - BCDE - CDEF - EFG$ of Exercise 9.1.
Algorithm 24 keeps the maximal clusters, weights edge $C_i - C_j$ of the complete cluster graph by $|C_i \cap C_j|$, and returns a maximum spanning tree (Theorem 9.9). The clusters are those of Exercise 9.1:

$$C_1 = ABD, \quad C_2 = EFG, \quad C_3 = BCDE, \quad C_4 = CDEF,$$
$$C_5 = DEF, \quad C_6 = EF, \quad C_7 = F.$$

Since $DEF, EF, F \subset CDEF$ and none of the other four contains another,

$$S = \{\, ABD,\ EFG,\ BCDE,\ CDEF \,\},$$

no reordering by Theorem 9.7 being needed here, as EO2JT2 wants only the set. The six weights $|C_i \cap C_j|$ are

edge	intersection	weight
$BCDE - CDEF$	CDE	3
$ABD - BCDE$	BD	2
$EFG - CDEF$	EF	2
$ABD - CDEF$	D	1
$EFG - BCDE$	E	1
$ABD - EFG$	$\emptyset$	0

A spanning tree on four nodes has three edges, so its weight is at most $3+2+2 = 7$; the three heaviest edges $BCDE-CDEF$, $ABD-BCDE$, $EFG-CDEF$ form a simple path, hence attain it. Omitting any of the three forces weight at most $3 + 2 + 1 = 6$, since the remaining weights are $\{1, 1, 0\}$, so the maximum spanning tree is unique:

$$ABD \ - \ BCDE \ - \ CDEF \ - \ EFG,$$

with separators BD, CDE, EF. This is Exercise 9.1's jointree; by Theorem 9.9 it has the jointree property, of width $4 - 1 = 3 = \text{width}(\pi, G)$.

Problem 9.3. Convert the dtree in Figure 9.28 to a jointree and then remove all nonmaximal clusters.
The dtree in Figure 9.28 is a full binary tree with thirteen nodes, numbered 1 to 13, whose leaves correspond to the seven families of the DAG of Figure 9.28 (nodes $A, \ldots, G$ with edges $A \to D$, $B \to D$, $B \to E$, $C \to E$, $C \to F$, $D \to F$, $E \to G$, $F \to G$). Its structure is

node	children	family (if a leaf)
1	$2, 3$	
2	$4, 5$	
3	—	C
4	$6, 7$	
5	—	EFG
6	—	CDF
7	$8, 9$	
8	—	BCE
9	$10, 11$	
10	—	B
11	$12, 13$	
12	—	ABD
13	—	A

Solution. Labelling each dtree node by cluster(T) gives a jointree, which reduces to the chain $ABD - BCDE - CDEF - EFG$; by Theorem 9.10 the dtree clusters satisfy the jointree property, and the leaf clusters are exactly the families.
Bottom-up, vars is the family at a leaf and $\text{vars}(T^l) \cup \text{vars}(T^r)$ at a nonleaf:

$$\begin{aligned}
\text{vars}(11) &= ABD \cup A = ABD,\\
\text{vars}(9) &= B \cup ABD = ABD,\\
\text{vars}(7) &= BCE \cup ABD = ABCDE,\\
\text{vars}(4) &= CDF \cup ABCDE = ABCDEF,\\
\text{vars}(2) &= ABCDEF \cup EFG = ABCDEFG,\\
\text{vars}(1) &= ABCDEFG \cup C = ABCDEFG.
\end{aligned}$$

Descending with $\text{cutset}(T) = (\text{vars}(T^l) \cap \text{vars}(T^r)) \setminus \text{acutset}(T)$ and $\text{context}(T) = \text{vars}(T) \cap \text{acutset}(T)$, where acutset($T$) unions the cutsets of the proper ancestors: $\text{cutset}(1) = C$, $\text{cutset}(2) = EF \setminus C = EF$, $\text{cutset}(4) = CD \setminus CEF = D$, $\text{cutset}(7) = B \setminus CDEF = B$, $\text{cutset}(9) = B \setminus BCDEF = \emptyset$, $\text{cutset}(11) = A \setminus BCDEF = A$. With $\text{cluster}(T) = \text{cutset}(T) \cup \text{context}(T)$ at nonleaves and vars(T) at leaves,

node	cutset	context	cluster
1	C	$\emptyset$	C
2	EF	C	CEF
3	—	C	C
4	D	CEF	$CDEF$
5	—	EF	EFG
6	—	CDF	CDF
7	B	CDE	$BCDE$
8	—	BCE	BCE
9	$\emptyset$	BD	BD
10	—	B	B
11	A	BD	ABD
12	—	ABD	ABD
13	—	A	A

a jointree of width $4 - 1 = 3$. Now apply the Section 9.4 transformations, both of which preserve the jointree properties: **remove cluster** deletes a C_j whose single neighbour C_i contains it, and **merge clusters** replaces neighbours by $C_i \cup C_j$, inheriting their neighbours. Removals: leaf 3 (C) into 1 (C); then leaf 1 (C) into 2 (CEF); then 2 (CEF) merges into 4 ($CDEF$), reattaching 5 to 4; leaf 6 (CDF) into 4; leaf 8 (BCE) into 7 ($BCDE$); 9 (BD) merges into 7, reattaching $10, 11$; leaf 10 (B) into 7; leaves 12 (ABD) and 13 (A) into 11 (ABD).
Surviving are 4, 5, 7, 11, with 5 and 7 on 4 and 11 on 7, and no remaining cluster contains another

($A \notin BCDE$, $G \notin CDEF$):

$$ABD \overset{BD}{—} BCDE \overset{CDE}{—} CDEF \overset{EF}{—} EFG,$$

the jointree of Exercises 9.1 and 9.2, still of width 3.

Problem 9.4. Construct a total elimination order that is consistent with the partial elimination order induced by the dtree in Figure 9.28. When the relative order of two variables is not fixed by the dtree, place them in alphabetic order.
The dtree in Figure 9.28 is the full binary tree on nodes $1, \ldots, 13$ in which node 1 has children 2 and 3; node 2 has children 4 and 5; node 4 has children 6 and 7; node 7 has children 8 and 9; node 9 has children 10 and 11; node 11 has children 12 and 13. The leaves carry the families of the DAG of Figure 9.28: node 3 is C, node 5 is EFG, node 6 is CDF, node 8 is BCE, node 10 is B, node 12 is ABD, node 13 is A.

Solution. $\pi = A, B, D, G, E, F, C$.
By Definition 9.11, X is eliminated at the node T with $X \in \text{cluster}(T) \setminus \text{context}(T)$, which at a nonleaf is $\text{cutset}(T)$, and by Theorem 9.3 this node is unique. From Exercise 9.3,

$$\begin{aligned} &\text{cutset}(1) = C, \quad \text{cutset}(2) = EF, \quad \text{cutset}(4) = D, \\ &\text{cutset}(7) = B, \quad \text{cutset}(9) = \emptyset, \quad \text{cutset}(11) = A, \end{aligned}$$

while at the leaves cluster $\setminus$ context is empty except at node 5, where it is $EFG \setminus EF = G$:

variable	eliminated at node
A	11
B	7
C	1
D	4
E	2
F	2
G	5

By Definition 9.12, $X < Y$ when X's node is a descendant of Y's, and the two are unordered when the nodes coincide. The dtree's ancestor chains are $11 \prec 9 \prec 7 \prec 4 \prec 2 \prec 1$ and $5 \prec 2 \prec 1$, with 5 and 11 in different subtrees of 2, so

$$A < B < D < E, \qquad A < B < D < F,$$

$$G < E, \qquad G < F, \qquad E < C, \qquad F < C,$$

with E, F unordered (both at node 2) and G unordered against A, B, D. Extending greedily, alphabetically first among the available minimal elements: A (over G), then B (over G), then D (over G), then G forced since E, F await it, then E before F alphabetically, then C:

$$\pi = A,\ B,\ D,\ G,\ E,\ F,\ C.$$

Theorem 9.4 predicts $\text{width}(\pi, G) \leq 3$, the Exercise 9.3 dtree width. Applying π to the moral graph (edges $A-B$, $A-D$, $B-C$, $B-D$, $B-E$, $C-D$, $C-E$, $C-F$, $D-F$, $E-F$, $E-G$, $F-G$) gives

$$ABD,\ BCDE,\ CDEF,\ EFG,\ CEF,\ CF,\ C,$$

the only fill-in being $D-E$ on eliminating B. The largest cluster has four variables, so $\text{width}(\pi, G) = 3$, matching the dtree width.

Problem 9.5. Construct a dtree for the DAG in Figure 9.28 using the elimination order $\pi = A, G, B, C, D, E, F$ and Algorithm 25. What is the width of this order? What is the width of the generated dtree?
The DAG of Figure 9.28 has nodes A, B, C, D, E, F, G and edges

$$A \to D, \quad B \to D, \quad B \to E, \quad C \to E,$$

$$C \to F, \quad D \to F, \quad E \to G, \quad F \to G,$$

so its seven families are A, B, C, ABD, BCE, CDF, EFG.

Solution. Both widths are 3.
The order: by Exercise 9.1, moralization adds $A-B$, $B-C$, $C-D$, $E-F$, and π induces the clusters $ABD, EFG, BCDE, CDEF, DEF, EF, F$, the largest of size four, so $\text{width}(\pi, G) = 3$ by Definition 9.5.
Algorithm 25 starts with $\Sigma = \{A, ABD, B, BCE, C, CDF, EFG\}$, one single-node dtree per family, and for each $\pi(i)$ composes all trees mentioning it (a single tree composes to itself, leaving Σ fixed). The trace:

$\pi(i)$	trees composed	result	Σ after
A	A, ABD	T_1, vars $= ABD$	T_1, B, BCE, C, CDF, EFG
G	EFG only	—	unchanged
B	T_1, B, BCE	T_3, vars $= ABCDE$	T_3, C, CDF, EFG
C	T_3, C, CDF	T_4, vars $= ABCDEF$	T_4, EFG
D	T_4 only	—	unchanged
E	T_4, EFG	T_6, vars $= ABCDEFG$	T_6
F	T_6 only	—	unchanged

so the algorithm returns T_6. Taking the left-linear COMPOSE, which may connect its arguments into a binary tree any way, in the order listed:

node	children
n_6	n_5, leaf EFG
n_5	n_4, leaf CDF
n_4	n_3, leaf C
n_3	n_2, leaf BCE
n_2	n_1, leaf B
n_1	leaf A, leaf ABD

rooted at n_6: a full binary tree whose seven leaves biject with the families, hence a dtree by Definition 9.14. Bottom-up,

$$\begin{aligned} &\text{vars}(n_1) = ABD, && \text{vars}(n_2) = ABD, && \text{vars}(n_3) = ABCDE, \\ &\text{vars}(n_4) = ABCDE, && \text{vars}(n_5) = ABCDEF, && \text{vars}(n_6) = ABCDEFG, \end{aligned}$$

then cutsets top-down:

$$\begin{aligned} \text{cutset}(n_6) &= (ABCDEF \cap EFG) \setminus \emptyset = EF, \\ \text{cutset}(n_5) &= (ABCDE \cap CDF) \setminus EF = CD, \\ \text{cutset}(n_4) &= (ABCDE \cap C) \setminus CDEF = \emptyset, \\ \text{cutset}(n_3) &= (ABD \cap BCE) \setminus CDEF = B, \\ \text{cutset}(n_2) &= (ABD \cap B) \setminus BCDEF = \emptyset, \\ \text{cutset}(n_1) &= (A \cap ABD) \setminus BCDEF = A. \end{aligned}$$

With $\text{context}(T) = \text{vars}(T) \cap \text{acutset}(T)$ and $\text{cluster}(T) = \text{cutset}(T) \cup \text{context}(T)$ at nonleaves, $\text{cluster}(T) = \text{vars}(T)$ at leaves, we get

node	cutset	context	cluster
n_6	EF	$\emptyset$	EF
n_5	CD	EF	$CDEF$
n_4	$\emptyset$	CDE	CDE
n_3	B	CDE	$BCDE$
n_2	$\emptyset$	BD	BD
n_1	A	BD	ABD
leaf A	—	A	A
leaf ABD	—	ABD	ABD
leaf B	—	B	B
leaf BCE	—	BCE	BCE
leaf C	—	C	C
leaf CDF	—	CDF	CDF
leaf EFG	—	EF	EFG

The largest clusters $CDEF$ and $BCDE$ have size four, so the dtree has width $4 - 1 = 3$, meeting the Theorem 9.14 bound $\text{width}(\pi, G) = 3$.

Problem 9.6. Consider the networks in Figure 6.6(d) and Figure 6.6(e) on Page 142. Show that the treewidth of these networks is 3. Hint: Use the preprocessing rules for generating optimal elimination prefixes.
Both networks have eleven nodes drawn in two columns. Name the node at the very top A; the left column, top to bottom, is B, D, F, H, J and the right column, top to bottom, is C, E, G, I, K, so that B and C sit on one row, D and E on the next, and so on.
The network of Figure 6.6(d) has the edges

$$A \to C, \quad B \to C, \quad C \to E, \quad E \to G, \quad G \to I, \quad I \to K,$$
$$B \to D, \quad D \to F, \quad F \to H, \quad H \to J, \quad F \to I, \quad C \to K.$$

That is: two parallel chains $B \to D \to F \to H \to J$ and $C \to E \to G \to I \to K$, the extra root $A \to C$, the cross edge $B \to C$, the cross edge $F \to I$, and the long edge $C \to K$.
The network of Figure 6.6(e) has all of these edges plus the two further cross edges

$$B \to E, \qquad D \to G.$$

Solution. Both reduce under the preprocessing rules to a K_4, so both have treewidth 3.
Work in the moral graphs G_d^m, G_e^m, widths with respect to a DAG being widths with respect to its moral graph (Definition 9.5). The multi-parent nodes are C (A, B), I (G, F), K (I, C) in G_d, joined by E (C, B) and G (E, D) in G_e, so moralization adds $A - B$, $F - G$, $C - I$, plus $D - E$ in G_e ($B - C$ is already there):

node	in G_d^m	in G_e^m
A	B, C	B, C
B	A, C, D	A, C, D, E
C	A, B, E, I, K	A, B, E, I, K
D	B, F	B, E, F, G
E	C, G	B, C, D, G
F	D, G, H, I	D, G, H, I
G	E, F, I	D, E, F, I
H	F, J	F, J
I	C, F, G, K	C, F, G, K
J	H	H
K	C, I	C, I

The Section 9.3.2 invariant is $\text{treewidth}(G) = \max(\text{treewidth}(G'), low)$, the eliminated variables forming an optimal prefix; start at $low = 1$, each graph having an edge. In both graphs the twig rule takes

J then H (degree 1, $low = 1$), and the simplicial rule takes A (neighbours B, C with $B - C$) and K (neighbours C, I with $C - I$), raising low to 2 and adding no fill-in.
(i) Figure 6.6(d). What remains on B, C, D, E, F, G, I is

$$\begin{array}{llll} B : C, D & C : B, E, I & D : B, F & E : C, G \\ F : D, G, I & G : E, F, I & I : C, F, G. & \end{array}$$

Here B, D, E have degree 2, hence are almost simplicial (a single neighbour is trivially a clique), so with $low = 2$ the series rule eliminates each, with fill-ins $C - D$, $C - F$, $C - G$ respectively. This leaves K_4 on $\{C, F, G, I\}$, of treewidth exactly 3 (at least 3 by Theorem 9.1, at most 3 since any order gives clusters $4, 3, 2, 1$). By the invariant,

$$\text{treewidth}(G_d^m) = \max(3, low) = \max(3, 2) = 3.$$

The prefix, completed by the four survivors, is

$$\pi_d = J,\ H,\ A,\ K,\ B,\ D,\ E,\ C,\ F,\ G,\ I,$$

whose induced cluster sequence is

$$\begin{array}{l} HJ,\ FH,\ ABC,\ CIK,\ BCD,\ CDF, \\ CEG,\ CFGI,\ FGI,\ GI,\ I, \end{array}$$

of maximum size 4, so $\text{width}(\pi_d, G_d) = 3$ and π_d is optimal.
(ii) Figure 6.6(e). After the same four eliminations the remaining G' on $\{B, C, D, E, F, G, I\}$ is

$$\begin{array}{lll} B : C, D, E & C : B, E, I & D : B, E, F, G \\ E : B, C, D, G & F : D, G, I & G : D, E, F, I \\ I : C, F, G. & & \end{array}$$

and the rules stall: every degree is ≥ 3 and no node is simplicial (B lacks $C - D$, C lacks $B - I$, and so on), while the almost simplicial rule needs $low \geq 3$. Raise the bound instead. Since J, H, A, K were simplicial when eliminated, no fill-in was ever added, so G' is the induced subgraph of G_e^m on $\{B, C, D, E, F, G, I\}$, with degrees

$$B \mapsto 3, \quad C \mapsto 3, \quad D \mapsto 4, \quad E \mapsto 4,$$

$$F \mapsto 3, \quad G \mapsto 4, \quad I \mapsto 3,$$

so G' has degree 3 in the sense of Definition 9.7. Degree lower-bounds treewidth (Exercise 9.12) and treewidth is subgraph-monotone, so $\text{treewidth}(G_e^m) \geq \text{treewidth}(G') \geq 3$ and we may set $low = 3$, which unlocks the almost simplicial rule at degree 3.
C (neighbours B, E, I) is almost simplicial: discarding I leaves the clique $\{B, E\}$, since $B - E$ is one of the two extra edges of Figure 6.6(e). Eliminating it adds fill-ins $B - I$, $E - I$:

$$\begin{array}{lll} B : D, E, I & D : B, E, F, G & E : B, D, G, I \\ F : D, G, I & G : D, E, F, I & I : B, E, F, G. \end{array}$$

Now B (neighbours D, E, I) is almost simplicial, $\{D, E\}$ being a clique; eliminating it adds $D - I$:

$$\begin{array}{lll} D : E, F, G, I & E : D, G, I & F : D, G, I \\ G : D, E, F, I & I : D, E, F, G. & \end{array}$$

Then E (neighbours D, G, I, pairwise adjacent) is simplicial of degree 3, so it goes with no fill-in and $low = 3$. The remainder on $\{D, F, G, I\}$ carries all six edges, again a K_4 of treewidth 3, so by the invariant

$$\text{treewidth}(G_e^m) = \max(3, low) = \max(3, 3) = 3.$$

The resulting order is

$$\pi_e = J,\ H,\ A,\ K,\ C,\ B,\ E,\ D,\ F,\ G,\ I,$$

with induced cluster sequence

$$HJ,\ FH,\ ABC,\ CIK,\ BCEI,\ BDEI,$$
$$DEGI,\ DFGI,\ FGI,\ GI,\ I,$$

whose largest clusters have four variables, so $\text{width}(\pi_e, G_e) = 3$.

Problem 9.7. Show that the leaf nodes of a Bayesian network represent a prefix for an optimal elimination order.
Here a leaf node of a Bayesian network is a node of its DAG that has no children, and a prefix τ of an elimination order π is a sequence of variables occurring at the beginning of π; τ is an optimal elimination prefix when it can be completed into an elimination order of width equal to the treewidth.

Solution. Every leaf of the DAG G is simplicial in the moral graph G^m, and distinct leaves are nonadjacent there, so the simplicial rule of Section 9.3.2 applies to them one after another; any enumeration $\tau = L_1, \ldots, L_k$ of the leaves is thus an optimal prefix. Widths with respect to G are widths with respect to G^m (Definition 9.5), so it suffices to argue in G^m.
Simpliciality: a leaf X has no children, so the DAG edges at X are exactly $U \to X$ for $U \in \text{Parents}(X)$, and X is never married as a co-parent; hence

$$\text{Neighbours}_{G^m}(X) = \text{Parents}(X),$$

which moralizing the family of X makes a clique. So X is simplicial of degree $d_X = |\text{Parents}(X)|$.
Nonadjacency: for leaves $X \neq Y$, neither is a parent of the other and neither is a co-parent of anything, so $X - Y \notin G^m$. Eliminating X therefore removes no neighbour of Y, and being simplicial it adds no fill-in at all, so Y is still simplicial with the same neighbourhood. Inductively, after $L_1, \ldots, L_{i-1}$ the node L_i remains simplicial with neighbourhood $\text{Parents}(L_i)$, no fill-in has appeared, the cluster induced at step i is the family $\{L_i\} \cup \text{Parents}(L_i)$, and the graph left is $G^m \setminus L$.
The simplicial rule of Section 9.3.2 is thus licensed at each of the k steps, its invariant giving both optimality of the prefix and

$$\text{treewidth}(G^m) = \max\Big(\max_i |\text{Parents}(L_i)|,\ \text{treewidth}(G^m \setminus L)\Big). \qquad \blacksquare$$

8.2 Exercises 9.8–9.14

Problem 9.8. Show that the root nodes of a Bayesian network, which have a single child each, represent a prefix for an optimal elimination order.

Solution. Each single-child root is simplicial in the moral graph G^m, so the simplicial rule of Section 9.3.2 applies to them in turn, exactly as for the leaves in Exercise 9.7. Widths with respect to G are widths with respect to G^m (Definitions 9.2 and 9.5).
Let X be a root whose only child is C. Its neighbours in G^m are its parents (none), its children (just C) and its spouses, the other parents of C, so

$$N_{G^m}(X) \ = \ \{C\} \cup \big(\text{Pa}(C) \setminus \{X\}\big),$$

a clique: moralization retains every edge $U \to C$ and marries all parents of C. Hence X is simplicial.
Let $X_1, \ldots, X_k$ be all such roots, in any order, and put $G^m_j = G^m - X_1 - \cdots - X_{j-1}$. Inductively, no fill-in has been added, so G^m_j is the induced subgraph of G^m on the survivors and

$$N_{G^m_j}(X_j) \ = \ N_{G^m}(X_j) \setminus \{X_1, \ldots, X_{j-1}\},$$

a subset of a G^m-clique and hence a clique of G_j^m, since deleting nodes destroys no edges among the survivors. So X_j is simplicial in G_j^m and its elimination again adds no fill-in.
Applying the simplicial rule k times therefore yields an optimal elimination order of G^m beginning $X_1, \ldots, X_k$, and the choice of order among them is immaterial by Theorem 9.2. ■

Problem 9.9. The preprocessing rules of Section 9.3.2 include the following two special cases of the Simplicial rule:
Islet rule Eliminate nodes with degree 0.
Twig rule Eliminate nodes with degree 1.
Show that the Islet and Twig rules are complete for graphs with treewidth 1; that is, they are sufficient to generate optimal elimination orders for such graphs.

Solution. A graph of treewidth ≤ 1 is a forest, on which one of the two rules always fires and never creates fill-in, so the order they generate has width ≤ 1 and is optimal.
Cycles force treewidth ≥ 2: let $Z_1 - \cdots - Z_m - Z_1$ be a cycle ($m \geq 3$), let π be any order, and let i be the first step eliminating a cycle node $X = \pi(i) = Z_t$, with distinct cycle neighbours $U = Z_{t-1}$, $W = Z_{t+1}$ still present in G_i. Elimination only adds edges and deletes the eliminated node (Definition 9.3), so $X - U$ and $X - W$ survive in G_i and

$$C_i \supseteq \{X, U, W\}, \qquad |C_i| \geq 3,$$

so width$(\pi, G) \geq 2$ (Definition 9.5) and, π being arbitrary, treewidth$(G) \geq 2$ (Definition 9.6). Contrapositively, treewidth$(G) \leq 1$ makes G a forest.
A nonempty forest has a node of degree ≤ 1: with no edges every degree is 0; otherwise an endpoint v of a longest path has no neighbour off the path (it would extend it) and none on the path but its predecessor (it would close a cycle), so $\deg(v) = 1$. One of the two rules therefore always fires, and eliminating a node of degree $d \leq 1$ creates no fill-in (no nonadjacent neighbour pair exists), leaving the forest $F - X$ and a cluster of size $d + 1 \leq 2$.
Iterating to the empty graph gives an order π with all clusters of size ≤ 2, so width$(\pi, G) \leq 1$. If G has an edge then treewidth$(G) \geq 1$ by Theorem 9.1, forcing width$(\pi, G) =$ treewidth$(G) = 1$ since no width falls below the treewidth; if G has none, every cluster is a singleton and both are 0. Either way π is optimal. ■

Problem 9.10. The preprocessing rules of Section 9.3.2 operate on an undirected graph while maintaining a value low that is a lower bound on the treewidth of the original graph. Three of the rules are:
Islet rule Eliminate nodes with degree 0.
Twig rule Eliminate nodes with degree 1.
Series rule Eliminate nodes with degree 2 if low ≥ 2.
Show that the Islet, Twig, and Series rules are complete for graphs with treewidth 2; that is, they are sufficient to generate optimal elimination orders for such graphs.

Solution. On a graph G with treewidth$(G) \leq 2$ the three rules never stall and the order they generate is optimal. Three ingredients.
(i) A nonempty graph has a node of degree at most its treewidth, since the degree of a graph lower-bounds its treewidth (Exercise 9.12); so a nonempty graph of treewidth ≤ 2 offers a node of degree 0, 1 or 2.
(ii) Eliminating a node X of degree $d \leq 2$ in H yields a minor H' (Definition 9.3).

- If $d \leq 1$ there is no nonadjacent pair of neighbors, so no fill-in is added and $H' = H - X$, a subgraph of H.
- If $d = 2$ with neighbors U, W, then H' is $H - X$ together with the single fill-in edge $U - W$ (absent when U and W are already adjacent, in which case $H' = H - X$). But H' is exactly the graph obtained from H by *contracting* the edge $X - U$: contraction replaces X and U by a node adjacent

to

$$\big(N(X) \cup N(U)\big) \setminus \{X, U\} \;=\; \{W\} \cup \big(N(U) \setminus \{X\}\big),$$

and renaming that node U gives precisely $H - X$ plus the edge $U - W$.
So H' is a minor of H (Definition 9.9), and since deleting nodes cannot raise the treewidth (restrict any order: every cluster shrinks) while contraction cannot either (Exercise 9.13),

$$\text{treewidth}(H') \;\leq\; \text{treewidth}(H).$$

(iii) Minimum degree ≥ 2 forces a cycle, hence treewidth ≥ 2 by Exercise 9.9: take a longest path, let v be an endpoint and u its predecessor; v has a neighbour $w \neq u$, which cannot lie off the path (it would extend it), so the path segment from w to v closes into a cycle of length ≥ 3.
Now run from $H_1 = G$ with any sound initial low. At each stage H_i is a minor of G, so treewidth$(H_i) \leq 2$ by (ii), and if nonempty it has a node of degree ≤ 2 by (i).

- If it has a node of degree 0 or 1, fire the Islet or Twig rule.
- Otherwise the minimum degree is exactly 2, so treewidth$(G) \geq$ treewidth$(H_i) \geq 2$ by (iii) and minor-monotonicity; low $\leftarrow 2$ is sound and the Series rule fires.

Either way one node goes, so the run terminates after n steps with an order π whose every cluster is $\{X\} \cup N_{H_i}(X)$ for some X of degree ≤ 2, of size ≤ 3; hence

$$\text{width}(\pi, G) \;\leq\; 2$$

by Definition 9.5. Two cases match this to the treewidth. (i) G a forest: every H_i is then a forest with a node of degree ≤ 1 (Exercise 9.9), so the Series rule never fires and the run is that of Exercise 9.9, giving width$(\pi, G) =$ treewidth$(G) \in \{0, 1\}$. (ii) G not a forest: it has a cycle, so treewidth$(G) \geq 2$ by Exercise 9.9, hence $= 2$, and as no width falls below the treewidth,

$$2 \;\leq\; \text{width}(\pi, G) \;\leq\; 2.$$

Either way π is optimal. ■

Problem 9.11. The min-degree elimination heuristic of Section 9.3.1 repeatedly eliminates a node having the smallest number of neighbors in the current graph. Show that the min-degree heuristic is optimal for graphs with treewidth ≤ 2; that is, on such graphs the order it generates has width equal to the treewidth.

Solution. Every graph in the min-degree run keeps treewidth ≤ 2, which caps every cluster at size 3.
Let π be a min-degree order on G with treewidth$(G) \leq 2$, with graph sequence $G_1 = G, \ldots, G_n$ and clusters C_i, and put $d_i = |C_i| - 1$, the minimum degree of G_i by the heuristic, that is degree(G_i) in the sense of Definition 9.7. Inductively, if treewidth$(G_i) \leq 2$ then $d_i \leq 2$ by Exercise 9.12, so eliminating $\pi(i)$ yields a minor of G_i by Exercise 9.10(ii) and treewidth$(G_{i+1}) \leq$ treewidth$(G_i) \leq 2$ by minor-monotonicity (Exercise 9.13 for the contraction case). Hence $|C_i| \leq 3$ throughout and, by Definition 9.5,

$$\text{width}(\pi, G) \;=\; \max_i |C_i| - 1 \;\leq\; 2.$$

Since width$(\pi, G) \geq$ treewidth(G) always (Definition 9.6), only a strict gap need be ruled out. (i) treewidth$(G) = 2$: the display forces equality. (ii) treewidth$(G) \leq 1$: then G is a forest (Exercise 9.9), whose minimum degree is ≤ 1, so each elimination adds no fill-in and leaves a forest; inductively $d_i \leq 1$ and width$(\pi, G) \leq 1$, which equals $1 =$ treewidth(G) by Theorem 9.1 if G has an edge and equals 0 otherwise. ■

Problem 9.12. Definition 9.7 defines the degree of a graph as the minimum number of neighbors attained by any of its nodes. Show that the degree of a graph is a lower bound on the treewidth of the graph.

Solution. The first cluster of any elimination order already witnesses the bound.
Let π be any elimination order for a nonempty G, with clusters $C_1, \ldots, C_n$ (Definition 9.4). Since $G_1 = G$ carries no fill-in yet,

$$C_1 \;=\; \{\pi(1)\} \cup N_G(\pi(1)), \qquad |C_1| \;=\; 1 + \deg_G(\pi(1)).$$

and the width of π is the largest cluster size minus one (Definition 9.5), hence at least $|C_1| - 1$:

$$\text{width}(\pi, G) \;\geq\; |C_1| - 1 \;=\; \deg_G(\pi(1)) \;\geq\; \min_{v \in V} \deg_G(v) \;=\; \text{degree}(G),$$

the last step because $\pi(1)$ is one of the nodes minimized over. As π was arbitrary, Definition 9.6 gives

$$\text{treewidth}(G) \;=\; \min_{\pi} \text{width}(\pi, G) \;\geq\; \text{degree}(G),$$

the required bound. Equivalently: every nonempty graph has a node of degree at most its treewidth, the form used in Exercises 9.10 and 9.11. ■

Problem 9.13. Exercise 9.25 defines a jointree for an undirected graph G as a pair (T, C), where T is a tree and C is a function mapping each node i of T to a label C_i, called a cluster, satisfying:

- The cluster C_i is a set of nodes in graph G.
- For every edge $X - Y$ in graph G, the variables X and Y appear together in some cluster C_i.
- The clusters of tree T satisfy the jointree property: if a node of G appears in two clusters C_i and C_j, it appears in every cluster on the path connecting i and j in T.

As in Definition 9.13, the width of a jointree is the size of its largest cluster minus one. Suppose that the treewidth of an undirected graph is the width of its best jointree (one with lowest width). Use this definition of treewidth to show that contracting an edge in a graph does not increase its treewidth. Here, contracting the edge $X - Y$ means replacing the nodes X and Y by a new node Z that is adjacent to the union of their neighbors, as illustrated in Figure 9.7.

Solution. Rename X and Y to Z inside every cluster of an optimal jointree; the same tree then works for the contracted graph, with no cluster larger.
Let G' be G with the edge $X - Y$ contracted, so its nodes are $(V(G) \setminus \{X, Y\}) \cup \{Z\}$, its edges being $A - B$ for edges of G avoiding X, Y, and $Z - A$ whenever $A \notin \{X, Y\}$ neighbours X or Y in G. Let (T, C) be a jointree for G of minimal width $w = \text{treewidth}(G)$ and define C' on the same T by

$$C'_i \;=\; \begin{cases} \big(C_i \setminus \{X, Y\}\big) \cup \{Z\}, & \text{if } C_i \cap \{X, Y\} \neq \emptyset, \\ C_i, & \text{otherwise.} \end{cases}$$

The three conditions hold for (T, C') over G'.
(1) Each C'_i drops X, Y and possibly adds Z, all of whose elements are nodes of G'.
(2) An edge $A - B$ of G' avoiding X, Y is an edge of G, so lies in some C_i, and both survive renaming; an edge $Z - A$ comes from an edge $X - A$ or $Y - A$ of G, whose cluster C_i renames to one containing Z and A.
(3) For $A \neq Z$ the renaming touches nothing, so $\{i : A \in C'_i\} = \{i : A \in C_i\}$ is connected. For Z, write

$$T_X = \{\, i : X \in C_i \,\}, \qquad T_Y = \{\, i : Y \in C_i \,\},$$

so that by construction

$$\{\, i : Z \in C'_i \,\} \;=\; \{\, i : C_i \cap \{X, Y\} \neq \emptyset \,\} \;=\; T_X \cup T_Y.$$

Each of T_X, T_Y is a connected subtree by the jointree property for (T, C), and they meet: $X - Y$ being an edge of G, some C_k holds both, so $k \in T_X \cap T_Y$. Two connected subtrees sharing a node have connected union – for $i \in T_X$, $j \in T_Y$, the i-to-k path stays in T_X and the k-to-j path in T_Y, and their concatenation contains the unique i-to-j path. So the jointree property holds for Z too. Finally, for each i,

$$|C'_i| \;=\; |C_i| - |C_i \cap \{X, Y\}| + \big[\, C_i \cap \{X, Y\} \neq \emptyset \,\big] \;\leq\; |C_i|,$$

the bracketed indicator being at most $|C_i \cap \{X, Y\}|$ when that intersection is nonempty and both sides vanishing otherwise. So $\text{width}(T, C') \leq w$, and as the treewidth of G' is the width of its best jointree,

$$\text{treewidth}(G') \;\leq\; w \;=\; \text{treewidth}(G). \qquad \blacksquare$$

Problem 9.14. Definition 9.7 defines the degree of a graph as the minimum number of neighbors attained by any of its nodes, and Definition 9.8 defines the degeneracy of a graph as the maximum degree attained by any of its subgraphs. Show that the degeneracy of a graph can be computed by generating a sequence of subgraphs, starting with the original graph, and then generating the next subgraph by removing a minimum-degree node. The degeneracy is then the maximum degree attained by any of the generated subgraphs.

Solution. The maximum M returned by the min-degree removal sequence equals $\text{degeneracy}(G)$. The procedure generates

$$G \;=\; H_1,\; H_2,\; \ldots,\; H_n, \qquad H_{i+1} \;=\; H_i - v_i,$$

where v_i is a node of minimum degree in H_i, and returns

$$M \;=\; \max_{1 \leq i \leq n} \text{degree}(H_i) \;=\; \max_{1 \leq i \leq n} \deg_{H_i}(v_i),$$

the two agreeing since v_i has minimum degree in H_i, and Definition 9.8 makes $\text{degeneracy}(G) = \max_H \text{degree}(H)$ over nonempty subgraphs.
The maximum is attained at an induced subgraph: a nonempty subgraph H with $S = V(H)$ sits inside $G[S]$, so $\deg_H(v) \leq \deg_{G[S]}(v)$ on S and

$$\text{degree}(H) \;\leq\; \text{degree}(G[S]).$$

so only the $G[S]$ matter.
$M \leq \text{degeneracy}(G)$: each $H_i = G[V(H_i)]$ is a nonempty subgraph of G, so $\text{degeneracy}(G) \geq \text{degree}(H_i)$ by Definition 9.8, and the maximum over i is M.
$M \geq \text{degeneracy}(G)$: fix $S \neq \emptyset$, put $\delta = \text{degree}(G[S])$, and let

$$i \;=\; \min\{\, j : v_j \in S \,\}$$

be the step removing the first node of S (every node is removed exactly once). By minimality $S \subseteq V(H_i)$, so $G[S]$ is an induced subgraph of H_i containing v_i and $\deg_{G[S]}(v_i) \leq \deg_{H_i}(v_i)$, restricting to S only dropping neighbours. Since v_i is one of the nodes δ minimizes over,

$$\delta \;=\; \text{degree}(G[S]) \;\leq\; \deg_{G[S]}(v_i) \;\leq\; \deg_{H_i}(v_i) \;=\; \text{degree}(H_i) \;\leq\; M.$$

the penultimate equality being precisely where the min-degree choice is used: v_i realizes $\text{degree}(H_i)$. Maximizing over S gives $\text{degeneracy}(G) \leq M$, so $M = \text{degeneracy}(G)$. $\blacksquare$

8.3 Exercises 9.15–9.21

Problem 9.15. Show the following about Algorithm 20, BFS OEO: at no point during the search can we have a node $(G_\rho, \rho, w_\rho, b_\rho)$ on the open list and another node $(G_{\rho'}, \rho', w_{\rho'}, b_{\rho'})$ on the closed list where $G_\rho = G_{\rho'}$.
Recall Algorithm 20, BFS OEO, which searches for an optimal elimination order of a graph G with n nodes. A search node is a tuple $(G_\tau, \tau, w_\tau, b_\tau)$: τ is an elimination prefix, G_τ is the subgraph obtained by applying τ to G, w_τ is the width of the prefix, and b_τ is a lower bound on $\text{treewidth}(G_\tau)$. The algorithm initializes the open list to $OL = \{(G, \tau, 0, b)\}$ with τ empty, and the closed list CL to empty. While OL is nonempty it removes from OL a node $(G_\tau, \tau, w_\tau, b_\tau)$ minimizing $\max(w_\tau, b_\tau)$; if G_τ is empty it returns τ; otherwise, for every variable X of G_τ it forms the child $(G_\rho, \rho, w_\rho, b_\rho)$

obtained by eliminating X, and then:
- if OL contains a node $(G_{\rho'}, \rho', w_{\rho'}, b_{\rho'})$ with $G_{\rho'} = G_\rho$, then that node is replaced by the new child when $w_\rho < w_{\rho'}$ and the new child is discarded otherwise;
- else if CL contains no node whose subgraph equals G_ρ, the child is added to OL;
- otherwise (a node with subgraph G_ρ sits on CL) the child is discarded.

Finally the expanded node $(G_\tau, \tau, w_\tau, b_\tau)$ is added to CL.

Solution. The claim is the second half of a loop invariant: (A) no two OL nodes share a subgraph; (B) no OL node shares a subgraph with a CL node. Both hold initially, OL being a singleton and CL empty. Note that a search node's subgraph depends only on the **set** of eliminated variables (Theorem 9.2), which is what makes the duplicate tests meaningful, and that eliminating a variable deletes exactly that variable (Definition 9.3), so a child ρ of τ has $|G_\rho| = |G_\tau| - 1$ and hence

$$G_\rho \neq G_\tau \quad \text{for every child } \rho \text{ of } \tau,$$

expansion being entered only for nonempty G_τ.
Assume (A) and (B) at the start of an iteration whose chosen node $(G_\tau, \tau, w_\tau, b_\tau)$ is expanded (otherwise the algorithm returns).
(A) survives: OL changes only by the replacement on Lines 18–19, which keeps the count of G_ρ-nodes at one and touches no other subgraph, and by the insertion on Line 22, reached only when Line 16 found no G_ρ-node on OL, raising that count from zero to one.
(B) survives insertion into OL: a child inserted on Line 19 has OL already carrying G_ρ, so (B) puts no G_ρ-node on CL; a child inserted on Line 22 has passed the Line 21 guard, which checks exactly that. These are the only ways into OL.
(B) survives insertion into CL on Line 25: the chosen node was the only OL node with subgraph G_τ by (A), and was removed on Line 6, while every node added to OL during the expansion is a child, with $G_\rho \neq G_\tau$ by the display. So no OL node carries G_τ when Line 25 runs.
Nodes never leave CL and no other line touches either list, so (A) and (B) persist. ■

Problem 9.16. Show that Algorithm 20, BFS OEO, will not choose a node $(G_\tau, \tau, w_\tau, b_\tau)$ from the open list where τ is a complete yet suboptimal elimination order.
Recall from Exercise 9.15 the structure of Algorithm 20: search nodes are tuples $(G_\tau, \tau, w_\tau, b_\tau)$ where τ is an elimination prefix of the input graph G, G_τ is the subgraph obtained by applying τ to G, w_τ is the width of the prefix, and b_τ is a lower bound on treewidth(G_τ). Each iteration removes from the open list a node minimizing $\max(w_\tau, b_\tau)$, returns τ if G_τ is empty, and otherwise generates one child per variable X of G_τ, with $w_\rho = \max(w_\tau, d)$ where d is the number of neighbors of X in G_τ; duplicates are resolved against the open and closed lists as described in Exercise 9.15, and the expanded node is moved to the closed list.

Solution. Every node the algorithm selects satisfies $f = \max(w_\sigma, b_\sigma) \leq w^\star = \text{treewidth}(G)$, so a selected complete order has $w_\tau \leq w^\star$ and is optimal.
Fix an optimal order $\pi = \pi(1), \ldots, \pi(n)$, and for $0 \leq k \leq n$ let H_k be the subgraph left by the prefix $\pi(1), \ldots, \pi(k)$, so $H_0 = G$ and H_n is empty.
Fact 1 (width decomposition). By Definition 9.4 the graph and cluster sequences induced by $\sigma\rho$ on G split into those of σ on G followed by those of ρ on G_σ, so by Definition 9.5

$$\text{width}(\sigma\rho, G) = \max\big(w_\sigma,\ \text{width}(\rho, G_\sigma)\big) \ \geq\ \max(w_\sigma, b_\sigma),$$

using $\text{width}(\rho, G_\sigma) \geq \text{treewidth}(G_\sigma) \geq b_\sigma$; thus f is admissible. Fact 2: $\text{treewidth}(H_k) \leq w^\star$, since the tail of π eliminates H_k within width $w^\star$.
Call $N = (G_\sigma, \sigma, w_\sigma, b_\sigma)$ **good** when $G_\sigma = H_k$ for some k and $w_\sigma \leq w^\star$. Then $\sigma\,\pi(k+1)\cdots\pi(n)$ is a complete order (Theorem 9.2) of width $\leq w^\star$ by Fact 1, and $f(N) \leq w^\star$ by Fact 2.
Fact 3 (persistence). Once $OL \cup CL$ holds a good node with subgraph H_k, it always does: a node leaves OL only by selection (going to CL) or by replacement with a same-subgraph node of strictly

smaller width, itself good; CL never loses nodes.
Induct on the iteration index t for (I1) the node selected at t has $f \le w^\star$, and (I2) OL holds a good node at the start of t. Initially $OL = \{(G, \varepsilon, 0, b)\}$, good since $G = H_0$, giving (I2) at $t = 1$; and (I2) gives (I1) at once, Line 6 minimizing f over a list containing a node of $f \le w^\star$.
Expanding a good node $N^\star = (H_k, \sigma, w_\sigma, b_\sigma)$ with H_k nonempty yields a good successor. Eliminating $\pi(k+1)$ from H_k gives subgraph H_{k+1} (Theorem 9.2) and width $w_\rho = \max(w_\sigma, d)$, where $\pi(k+1)$ together with its d neighbours in H_k is exactly the cluster C_{k+1} induced by π on G, so $d = |C_{k+1}| - 1 \le w^\star$ and the child is good. Lines 16–23 then either update the OL node for H_{k+1} to width $\min(w_\rho, w_{\rho'}) \le w^\star$, or insert the good child, or discard it because CL holds an H_{k+1}-node, which was selected earlier and so has width $\le f \le w^\star$ by (I1). All three leave a good H_{k+1}-node in $OL \cup CL$.
For (I2) at $t+1$, assuming the algorithm did not return at t, put

$$J = \{\, k : OL \cup CL \text{ contains a good node with subgraph } H_k \,\},$$

at the start of iteration $t+1$. Then $0 \in J$ by Fact 3, so $k = \max J$ exists; let M be a good H_k-node in $OL \cup CL$. Were $M \in CL$, Exercise 9.15 would put no H_k-node at all on OL; also $k < n$, an H_n-node never reaching CL since selecting it returns. But M reached CL by being selected and expanded while good (width $\le f \le w^\star$ by (I1)), so by the previous paragraph and Fact 3 a good H_{k+1}-node survives, contradicting maximality of k. Hence $M \in OL$, which is (I2).
Now let a selected node have τ complete, so G_τ is empty and $b_\tau \le 0 \le w_\tau$, giving $f = w_\tau \le w^\star$ by (I1). Since $w_\tau = \text{width}(\tau, G) \ge \text{treewidth}(G) = w^\star$ (Definition 9.6), $w_\tau = w^\star$ and τ is optimal. ■

Problem 9.17. Prove the correctness of Algorithm 21, JT2EO. One way to prove this is to bound the size of factors constructed by the elimination Algorithm VE PR from Chapter 6 while using the elimination order generated by JT2EO.
Algorithm 21, JT2EO, takes a jointree $(T, \mathbf{C})$ for a DAG G (Definition 9.13) and returns an elimination order as follows. It starts with an empty order π; while tree T has more than one node, it removes a node i that has a single neighbor j and appends the variables $C_i \setminus (C_i \cap C_j)$ to π (in any order); when a single node r remains, it appends the variables of C_r to π and returns π. Correctness means: π is an elimination order for G and $\text{width}(\pi, G)$ is no greater than the width of the jointree, where the width of a jointree is the size of its largest cluster minus one, and $\text{width}(\pi, G)$ is the width of π with respect to the moral graph G^m of G (Definitions 9.1, 9.2, and 9.5).

Solution. Every factor VE PR builds under the order π returned by JT2EO lives inside a jointree cluster, so $\text{width}(\pi, G) \le w$, the jointree width.
JT2EO removes leaves one at a time, so the remaining tree is always a subtree of T, and since a path in a subtree is the path in T, the jointree property (Definition 9.13) holds at every stage.
π is an elimination order. Let $T_X = \{i : X \in C_i\}$, a connected subtree. When JT2EO removes $i \in T_X$ with sole neighbour j: if some other remaining node k lies in T_X, the path from i to k starts with $i - j$, so $X \in C_j$ and X is not appended; otherwise $X \notin C_j$ and X is appended now. So X is appended exactly once, when the last node of T_X goes (or with C_r if that node is the survivor r). Every variable lies in its own family, hence in some cluster (Definition 9.13), so π lists all variables exactly once.
The width claim reduces to a bound on VE factors. Run VE PR on any network with DAG G using π, with empty query and evidence, so the initial factors are the CPTs. By Definition 6.4, their interaction graph is the moral graph G^m, since $\text{vars}(f_X)$ is the family of X and making each family a clique is moralization (Definition 9.1). The two observations after Definition 6.4 say that eliminating $\pi(i)$ builds a factor over $\pi(i)$ and its current neighbours, and that the new interaction graph connects those neighbours pairwise and deletes $\pi(i)$ – exactly graph elimination (Definition 9.3). Inductively the interaction graph before step i is G_i and the factor built has variable set C_i^π, so

$$\text{width}(\pi, G) \;=\; \max_i |C_i^\pi| - 1 \;=\; \big(\text{largest number of variables in a constructed factor}\big) - 1,$$

and it suffices to confine every constructed factor to a cluster.
Every family lies in some cluster (Definition 9.13), so assign each CPT f_X to a node i with $\text{vars}(f_X) \subseteq C_i$, and run JT2EO and VE PR in lockstep under the invariant

(INV) every factor held at a node i of the remaining tree has its variables inside C_i,
which holds initially. Let JT2EO remove leaf i with sole neighbour j, so VE PR eliminates $V = C_i \setminus (C_i \cap C_j)$ in some order, and let $X \in V$. No remaining cluster but C_i contains X, so by (INV) no factor away from i mentions X, and eliminating X multiplies exactly the factors at i, building a factor inside C_i and leaving $\sum_X \prod f_k$ inside $C_i \setminus \{X\}$ at node i. The argument repeats down V, and the surviving factor lands inside $C_i \setminus V = C_i \cap C_j \subseteq C_j$, so moving it to j restores (INV). When only r remains, all factors lie inside C_r and eliminating C_r builds only subsets of it.
Hence every constructed factor has at most $|C_i| \leq w + 1$ variables, so every C_i^π does, and

$$\text{width}(\pi, G) \;=\; \max_i |C_i^\pi| - 1 \;\leq\; w. \qquad \blacksquare$$

Problem 9.18. Consider a cluster sequence $C_1, \ldots, C_n$ induced by an elimination order π on graph G. Show that it is impossible for a cluster C_i to be contained in another cluster C_j where $j > i$.
Recall Definition 9.4: eliminating nodes from G according to π induces a graph sequence $G_1, G_2, \ldots, G_n$ with $G_1 = G$ and G_{i+1} the result of eliminating node $\pi(i)$ from G_i, and a cluster sequence $C_1, \ldots, C_n$ where C_i consists of node $\pi(i)$ together with its neighbors in G_i.

Solution. The witness is $\pi(i)$ itself, which lies in C_i but is gone by G_j.
The nodes of G_k are those of G other than $\pi(1), \ldots, \pi(k-1)$: eliminating a node adds fill-in edges and deletes that node (Definition 9.3), leaving the node set otherwise untouched, so the claim follows by induction from $G_1 = G$.
Fix $i < j$. Then $\pi(i) \in C_i$ by Definition 9.4, while C_j consists of nodes of G_j, from which $\pi(i)$ has already been deleted since $i \leq j - 1$. Hence

$$\pi(i) \in C_i \quad \text{and} \quad \pi(i) \notin C_j,$$

so $C_i \not\subseteq C_j$. $\blacksquare$

Problem 9.19. Prove Lemma 9.5.
Lemma 9.5 reads: consider a dtree node T and define $\text{vars}^\uparrow(T) = \emptyset$ if T is a root node; otherwise $\text{vars}^\uparrow(T) = \bigcup_{T'} \text{vars}(T')$, where T' ranges over the leaf dtree nodes connected to node T through its parent. We have

$$\begin{aligned}
\text{cutset}(T) &= \big(\text{vars}(T^l) \cap \text{vars}(T^r)\big) \setminus \text{vars}^\uparrow(T), \\
\text{context}(T) &= \text{vars}(T) \cap \text{vars}^\uparrow(T), \\
\text{cluster}(T) &= \big(\text{vars}(T^l) \cap \text{vars}(T^r)\big) \cup \big(\text{vars}(T^l) \cap \text{vars}^\uparrow(T)\big) \cup \big(\text{vars}(T^r) \cap \text{vars}^\uparrow(T)\big),
\end{aligned}$$

the first and third statements being about nonleaf nodes T. This implies that $X \in \text{cluster}(T)$ only if $X \in \text{vars}(T)$.
Recall the definitions of Section 9.5 for a dtree, a full binary tree whose leaves correspond to the families of a DAG: $\text{vars}(T) = \text{vars}(T^l) \cup \text{vars}(T^r)$ for nonleaf T; $\text{cutset}(T) = (\text{vars}(T^l) \cap \text{vars}(T^r)) \setminus \text{acutset}(T)$ for nonleaf T; $\text{acutset}(T) = \bigcup_{T^\star} \text{cutset}(T^\star)$ over the proper ancestors $T^\star$ of T; $\text{context}(T) = \text{vars}(T) \cap \text{acutset}(T)$; and $\text{cluster}(T) = \text{vars}(T)$ for a leaf, $\text{cutset}(T) \cup \text{context}(T)$ otherwise.

Solution. Everything follows from $\text{acutset}(T) \subseteq U(T)$ and $\text{vars}(T) \cap U(T) \subseteq \text{acutset}(T)$, where $U(T) = \text{vars}^\uparrow(T)$.
Note that $U(T)$ unions vars over the leaves **outside** the subtree of T (deleting T leaves exactly the part hanging off its parent), that $\text{vars}(T)$ unions vars over the leaves inside it, and that $\text{vars}(T) \subseteq \text{vars}(T^p)$.
First inclusion: if $X \in \text{cutset}(T^\star) \subseteq \text{vars}(T^{\star l}) \cap \text{vars}(T^{\star r})$ for a proper ancestor $T^\star$, with T inside the subtree of, say, $T^{\star l}$, then X occurs at a leaf under $T^{\star r}$, which is outside the subtree of T; so $X \in U(T)$.
Second inclusion: let $X \in \text{vars}(T) \cap U(T)$, occurring at a leaf L outside the subtree of T, and let $T^\star$ be the deepest node whose subtree holds both. Then $T^\star \neq T$ is a proper ancestor of T, with T and

L under different children, say $T^{\star l}$ and $T^{\star r}$, so upward monotonicity gives $X \in \mathrm{vars}(T^{\star l}) \cap \mathrm{vars}(T^{\star r})$. Either $X \in \mathrm{acutset}(T^\star)$, so X lies in the cutset of an ancestor of T, or $X \in \mathrm{cutset}(T^\star)$ by definition of cutset; either way $X \in \mathrm{acutset}(T)$.
Hence $\mathrm{vars}(T) \cap \mathrm{acutset}(T)$, $\mathrm{vars}(T) \cap U(T)$ and $\mathrm{vars}(T) \cap \mathrm{acutset}(T)$ chain into each other, so all coincide:

$$\mathrm{context}(T) = \mathrm{vars}(T) \cap \mathrm{acutset}(T) = \mathrm{vars}(T) \cap U(T).$$

For the cutset, take T nonleaf and abbreviate $L = \mathrm{vars}(T^l)$, $R = \mathrm{vars}(T^r)$, $U = U(T)$, so $\mathrm{vars}(T) = L \cup R \supseteq L \cap R$. Subtracting a set from a subset of $\mathrm{vars}(T)$ is subtracting its intersection with $\mathrm{vars}(T)$, so

$$\begin{aligned}\mathrm{cutset}(T) &= (L \cap R) \setminus \mathrm{acutset}(T) \\ &= (L \cap R) \setminus \big(\mathrm{vars}(T) \cap \mathrm{acutset}(T)\big) \\ &= (L \cap R) \setminus \big(\mathrm{vars}(T) \cap U\big) \\ &= (L \cap R) \setminus U,\end{aligned}$$

the third equality being the context equation and the last using $L \cap R \subseteq \mathrm{vars}(T)$. For the cluster, again with T nonleaf,

$$\mathrm{cluster}(T) = \mathrm{cutset}(T) \cup \mathrm{context}(T) = \big((L \cap R) \setminus U\big) \cup \big((L \cup R) \cap U\big),$$

so, distributing the second term,

$$\mathrm{cluster}(T) = \big((L \cap R) \setminus U\big) \cup (L \cap U) \cup (R \cap U).$$

which equals $(L \cap R) \cup (L \cap U) \cup (R \cap U)$: $\subseteq$ is clear, and for $\supseteq$ only $L \cap R$ needs covering, an X there lying in $(L \cap R) \setminus U$ when $X \notin U$ and in $L \cap U$ otherwise.
Finally $\mathrm{cluster}(T) \subseteq \mathrm{vars}(T)$, trivially at a leaf and because all three terms sit inside $L \cup R$ at a nonleaf. ■

Problem 9.20. Consider a dtree generated from an elimination order π using Algorithm 25. Is it possible for the order not to be consistent with the dtree according to Definition 9.12, that is, the total order π is not compatible with the partial order induced by the dtree? Either show impossibility or provide a concrete example showing such a possibility.
Algorithm 25, EO2DT(G, π), works as follows. It initializes a collection Σ of one-node dtrees, one per family of the DAG G. Then, for $i = 1$ to the length of π, it collects the trees $T_1, \ldots, T_m$ in Σ that contain variable $\pi(i)$, removes them from Σ, and adds $T = \mathrm{COMPOSE}(T_1, \ldots, T_m)$ to Σ; the COMPOSE operator connects the given binary trees arbitrarily into a single binary tree. It finally returns the composition of all trees remaining in Σ.
Definition 9.11 says that a variable X is eliminated at dtree node T precisely when $X \in \mathrm{cluster}(T) \setminus \mathrm{context}(T)$, and Theorem 9.3 says each variable is eliminated at a unique node. Definition 9.12 says the dtree induces the partial order in which $X < Y$ when the node at which X is eliminated is a descendant of the node at which Y is eliminated, and $X = Y$ when the two are eliminated at the same node. A total order is consistent with the dtree when $X < Y$ in this partial order implies that X precedes Y in the total order.

Solution. Yes. Take G on X, Y, Z, W with edges $X \to Y$, $X \to Z$, $Y \to Z$, $X \to W$, families

$$L_1 = \{X\}, \quad L_2 = \{X, Y\}, \quad L_3 = \{X, Y, Z\}, \quad L_4 = \{X, W\},$$

and $\pi = X, Y, Z, W$. At $i = 1$ the variable X occurs in all four trees, so EO2DT replaces them by $\mathrm{COMPOSE}(L_1, L_2, L_3, L_4)$, which may be the vine

$$N_1 = (L_2, L_3), \qquad N_2 = (N_1, L_4), \qquad N_3 = (N_2, L_1),$$

rooted at N_3; the later steps see a single tree and change nothing. With $\mathrm{vars}(N_1) = XYZ$ and $\mathrm{vars}(N_2) = \mathrm{vars}(N_3) = XYZW$, the Section 9.5 definitions give

node	vars	cutset	context	cluster	eliminated here
N_3	XYZW	X		X	X
N_2	XYZW		X	X	
N_1	XYZ	Y	X	XY	Y
L_1	X		X	X	
L_2	XY		XY	XY	
L_3	XYZ		XY	XYZ	Z
L_4	XW		X	XW	W

The two loaded rows: $\text{acutset}(N_3) = \emptyset$ and $\text{cutset}(N_3) = XYZW \cap X = X$, so X is eliminated at N_3; and since $\text{cutset}(N_2) = (XYZ \cap XW) \setminus X = \emptyset$ gives $\text{acutset}(N_1) = X$,

$$\text{cutset}(N_1) = (\text{vars}(L_2) \cap \text{vars}(L_3)) \setminus X = (XY \cap XYZ) \setminus X = Y,$$

so Y is eliminated at N_1, each variable at a unique node as Theorem 9.3 requires. (The remaining rows: Check!)

But N_1 is a descendant of N_3, so Definition 9.12 imposes $Y < X$, while π places X first. Hence π is inconsistent with the dtree EO2DT built from it.

Problem 9.21. Provide a polytime, width-preserving algorithm for converting a jointree into a dtree. Your algorithm should be direct, bypassing the notion of an elimination order.

Recall Definition 9.13: a jointree for a DAG G is a pair $(T, \mathbf{C})$ where T is a tree and C_i is a cluster of nodes of G for each tree node i, such that every family of G appears in some cluster and, if a node of G appears in two clusters C_i and C_j, it appears in every cluster on the path between i and j (the jointree property); the width of the jointree is the size of its largest cluster minus one.

Recall Definition 9.14: a dtree for G is a pair (T, vars) where T is a full binary tree whose leaves are in one-to-one correspondence with the families of G, with $\text{vars}(L)$ the corresponding family; vars, cutsets, contexts, and clusters extend to internal nodes as in Section 9.5, and the width of a dtree is the size of its largest cluster minus one. Width-preserving means the output dtree has width no greater than that of the input jointree.

Solution. JT2DT: root the jointree, assign each family to a cluster containing it, and compose each node's families and child dtrees into a binary tree bottom-up. Let $(T, \mathbf{C})$ have width w, so $|C_i| \leq w+1$.

1. Root the jointree at an arbitrary node r, orienting each edge away from r.
2. Assign every family $\mathbf{F}$ of G to a single tree node $\text{node}(\mathbf{F})$ with $\mathbf{F} \subseteq C_{\text{node}(\mathbf{F})}$. Such a node exists by Condition 2 of Definition 9.13; if several qualify, pick any one. Let $\text{fams}(i)$ be the set of families assigned to node i.
3. Discard every rooted subtree that contains no assigned family (such subtrees carry no information); this does not touch the root, since G has at least one family.
4. Process the remaining nodes bottom-up. For node i with children $i_1, \ldots, i_k$ (after step 3) and assigned families $\mathbf{F}_1, \ldots, \mathbf{F}_m$, let the *components* of i be the fresh dtree leaves $L_1, \ldots, L_m$ with $\text{vars}(L_j) = \mathbf{F}_j$, together with the already-built dtrees $D(i_1), \ldots, D(i_k)$. Set
   - $D(i)$ = the unique component, if there is exactly one;
   - $D(i)$ = any full binary tree whose leaves are exactly the components – for instance the vine $((c_1, c_2), c_3), \ldots)$ – if there are two or more.
5. Return $D(r)$.

Step 3 leaves every surviving node at least one component, so $D(i)$ is defined; each jointree node and family is touched once, giving linear time and no elimination order. The output is a dtree by Definition 9.14: leaves biject with families, and every Step 4 node has exactly two children.

For the width, set, for a surviving node i,

$$\mathbf{V}_i = \bigcup \{\, \mathbf{F} : \text{node}(\mathbf{F}) \text{ lies in the subtree rooted at } i \,\},$$

$$\mathbf{U}_i = \bigcup \{\, \mathbf{F} : \text{node}(\mathbf{F}) \text{ lies outside the subtree rooted at } i \,\}.$$

so that $\text{vars}(D(i)) = \mathbf{V}_i$ and the final-dtree leaves outside $D(i)$ carry exactly $\mathbf{U}_i$. The jointree property gives two facts, both because every path joining a node inside the subtree of i to one outside (or joining subtrees of two distinct children) passes through i, while each assigned family sits inside its node's cluster:

- (A) $X \in \mathbf{V}_{i_a} \cap \mathbf{V}_{i_b}$ for distinct children i_a, i_b implies $X \in C_i$;
- (B) $X \in \mathbf{V}_i \cap \mathbf{U}_i$ implies $X \in C_i$.

Now every dtree cluster sits inside a jointree cluster. For a leaf L_j built from $\mathbf{F}_j$ assigned to i, $\text{cluster}(L_j) = \mathbf{F}_j \subseteq C_i$. For an internal $T^\star$ built while composing the components of i, with $U = \text{vars}^\uparrow(T^\star)$, Lemma 9.5 gives

$$\text{cluster}(T^\star) \subseteq \big(\text{vars}(T^{\star l}) \cap \text{vars}(T^{\star r})\big) \cup \big(\text{vars}(T^{\star l}) \cap U\big) \cup \big(\text{vars}(T^{\star r}) \cap U\big).$$

Now $\text{vars}(T^{\star l})$ and $\text{vars}(T^{\star r})$ union the vars of two disjoint sets of components of i, while U unions those of the remaining components together with $\mathbf{U}_i$. So $X \in \text{cluster}(T^\star)$ lies in at least two distinct members of

$$\{\, \mathbf{F}_1, \ldots, \mathbf{F}_m,\ \mathbf{V}_{i_1}, \ldots, \mathbf{V}_{i_k},\ \mathbf{U}_i \,\}.$$

and in each case $X \in C_i$: (i) if one member is a family $\mathbf{F}_j$ assigned to i, then $X \in \mathbf{F}_j \subseteq C_i$; (ii) if they are $\mathbf{V}_{i_a}$, $\mathbf{V}_{i_b}$ with $a \neq b$, by (A); (iii) if they are $\mathbf{V}_{i_a}$, $\mathbf{U}_i$, by (B). These exhaust the cases, $\mathbf{U}_i$ occurring once. So $\text{cluster}(T^\star) \subseteq C_i$.

Every dtree cluster therefore has at most $w + 1$ variables, and the dtree has width at most w. ■

8.4 Exercises 9.22–9.28

Problem 9.22. Show the following. Given a probability distribution $\Pr(\mathbf{X})$ that is induced by a Bayesian network and given a corresponding dtree, the distribution can be expressed in the following form:

$$\Pr(\mathbf{X}) \;=\; \frac{\displaystyle\prod_{\text{dtree node } T} \Pr(\text{cluster}(T))}{\displaystyle\prod_{\text{dtree node } T} \Pr(\text{context}(T))}.$$

Both products range over all nodes T of the dtree, leaves included.

Solution. The identity is Theorem 9.12 applied to the dtree labelled by its clusters, once separators are recognized as contexts.

$(T, \text{cluster})$ is a jointree for G by Definition 9.13: each cluster is a set of variables of G; each family appears as $\text{cluster}(L) = \text{vars}(L)$ at its leaf, the leaves biject with the families; and the jointree property is Theorem 9.10. Hence Theorem 9.12 gives

$$\Pr(\mathbf{X}) \;=\; \frac{\displaystyle\prod_{\text{jointree node } i} \Pr(C_i)}{\displaystyle\prod_{\text{jointree edge } i-j} \Pr(C_i \cap C_j)}.$$

whose numerator is already $\prod_T \Pr(\text{cluster}(T))$. In a rooted tree the map $T \mapsto T - T^p$ bijects the non-root nodes with the edges, and the separator of $T - T^p$ is $\text{cluster}(T) \cap \text{cluster}(T^p) = \text{context}(T)$ by Theorem 9.13(e), so

$$\prod_{\text{jointree edge } i-j} \Pr(C_i \cap C_j) \;=\; \prod_{T \neq \text{root}} \Pr(\text{context}(T)).$$

The root R has no ancestors, so $\text{acutset}(R) = \varnothing$, whence $\text{context}(R) = \varnothing$ and $\Pr(\text{context}(R)) = 1$, the marginal over no variables. Multiplying the denominator by it changes nothing:

$$\prod_{T \neq \text{root}} \Pr(\text{context}(T)) \;=\; \prod_{\text{dtree node } T} \Pr(\text{context}(T)).$$

Substituting both into the jointree factorization yields

$$\Pr(\mathbf{X}) \;=\; \frac{\displaystyle\prod_{\text{dtree node } T} \Pr(\text{cluster}(T))}{\displaystyle\prod_{\text{dtree node } T} \Pr(\text{context}(T))},$$

which is the required identity. ■

For the dtree of Figure 9.28, the clusters and contexts are

node	cluster	context
1	C	$\varnothing$
2	CEF	C
3	C	C
4	$CDEF$	CEF
5	EFG	EF
6	CDF	CDF
7	$BCDE$	CDE
8	BCE	BCE
9	BD	BD
10	B	B
11	ABD	BD
12	ABD	ABD
13	A	A

so the identity reads

$$\begin{aligned}\Pr(A,\ldots,G) = {} & \frac{\Pr(C)\Pr(CEF)\Pr(C)\Pr(CDEF)\Pr(EFG)\Pr(CDF)\Pr(BCDE)}{\Pr(C)\Pr(C)\Pr(CEF)\Pr(EF)\Pr(CDF)\Pr(CDE)\Pr(BCE)} \\ & \times \frac{\Pr(BCE)\Pr(BD)\Pr(B)\Pr(ABD)\Pr(ABD)\Pr(A)}{\Pr(BD)\Pr(B)\Pr(BD)\Pr(ABD)\Pr(A)}.\end{aligned}$$

in which every nonmaximal cluster cancels against an equal context, leaving

$$\Pr(A,\ldots,G) = \frac{\Pr(CDEF)\ \Pr(BCDE)\ \Pr(ABD)\ \Pr(EFG)}{\Pr(CDE)\ \Pr(BD)\ \Pr(EF)},$$

the Theorem 9.12 factorization for the minimal jointree of Exercise 9.3.

Problem 9.23. Show that a dtree will have width $\le 4w+1$ if each of its nodes T satisfies the following conditions:

- The cutset of T has no more than $w+1$ variables.
- No more than two thirds of the variables in context(T) appear in either vars(T^l) or vars(T^r).

Here w is a nonnegative integer, T^l and T^r are the left and right children of T, and the second condition is to be read as $|\text{context}(T) \cap \text{vars}(T^l)| \le \frac{2}{3}|\text{context}(T)|$ and $|\text{context}(T) \cap \text{vars}(T^r)| \le \frac{2}{3}|\text{context}(T)|$.

Solution. Every cluster satisfies $|\text{cluster}(T)| \le (w+1)+(3w+1) = 4w+2$, so the width, one less than the largest cluster, is at most $4w+1$. Cutsets at leaves are read as $\text{cutset}(L) = \text{vars}(L) \setminus \text{acutset}(L)$, which is what makes the first hypothesis a condition on every node; with that reading $\text{cluster}(T) = \text{cutset}(T) \uplus \text{context}(T)$ at every node (Theorem 9.13(a) at a nonleaf; at a leaf the two sets partition vars(L)), so

$$|\text{cluster}(T)| = |\text{cutset}(T)| + |\text{context}(T)|.$$

Since $|\text{cutset}(T)| \le w+1$ by the first hypothesis, only $|\text{context}(T)| \le 3w+1$ is at issue.

Inheritance. For a child T^c of a nonleaf T we have $\text{acutset}(T^c) = \text{acutset}(T) \cup \text{cutset}(T)$; intersecting with vars$(T^c)$ and using vars$(T^c) \subseteq$ vars(T) on the first part and $\text{cutset}(T) \subseteq \text{vars}(T^l) \cap \text{vars}(T^r)$ on

the second,

$$\operatorname{context}(T^c) = \big(\operatorname{context}(T) \cap \operatorname{vars}(T^c)\big) \uplus \operatorname{cutset}(T),$$

disjointly because $\operatorname{context}(T) \cap \operatorname{cutset}(T) = \varnothing$ by Theorem 9.13(a).
Induction on depth. At the root acutset $= \varnothing$, so $|\text{context}| = 0$. Given $|\operatorname{context}(T)| \le 3w+1$, the second hypothesis yields $|\operatorname{context}(T) \cap \operatorname{vars}(T^c)| \le \frac{2}{3}(3w+1) = 2w + \frac{2}{3}$, hence $\le 2w$, a cardinality being an integer and $2w$ being one too; so by the inheritance identity and the first hypothesis

$$|\operatorname{context}(T^c)| \le 2w + (w+1) = 3w+1. \qquad \blacksquare$$

Problem 9.24. Use Algorithm BAL DT of Section 9.5.3 to balance the dtree in Figure 9.28.
Figure 9.28 shows a Bayesian network over the variables $A, \ldots, G$ with edges

$$A \to D, \quad B \to D, \quad B \to E, \quad C \to E,$$
$$C \to F, \quad D \to F, \quad E \to G, \quad F \to G,$$

so that its families are A, B, C, ABD, BCE, CDF, EFG.
The figure also shows a corresponding dtree whose thirteen nodes are numbered $1, \ldots, 13$ and connected as follows (a caterpillar: leaves hang off the single long path $1, 2, 4, 7, 9, 11$):

- node 1 (the root) has children 2 and 3;
- node 2 has children 4 and 5;
- node 4 has children 6 and 7;
- node 7 has children 8 and 9;
- node 9 has children 10 and 11;
- node 11 has children 12 and 13.

The seven leaves carry the families

$$\operatorname{vars}(3) = C, \;\; \operatorname{vars}(5) = EFG, \;\; \operatorname{vars}(6) = CDF, \;\; \operatorname{vars}(8) = BCE,$$

$$\operatorname{vars}(10) = B, \;\; \operatorname{vars}(12) = ABD, \;\; \operatorname{vars}(13) = A.$$

Recall BAL DT. Label every nonleaf node of the dtree with the empty dtree and every leaf node with itself; take the label-combining operation to be COMPOSE. Then apply tree contraction repeatedly until a single node remains and return its label. One contraction step is a **rake** (absorb every leaf into its parent) followed by a **compress** (identify every maximal chain $N_1, \ldots, N_k$ and absorb N_i into N_{i+1} for odd i), where $N_1, \ldots, N_k$ is a chain when N_{i+1} is the only child of N_i for $1 \le i < k$ and N_k has exactly one child and that child is not a leaf.

Solution. BAL DT halts after four contraction steps and returns the dtree displayed below. Write $\langle t_1, t_2 \rangle$ for what COMPOSE returns on two nonempty arguments; on the empty dtree it returns the other argument. Nonleaves start labelled empty, leaves with themselves.
(i) *Rake* the leaves $3, 5, 6, 8, 10, 12, 13$ into their parents:

$$1 \leftarrow C, \;\; 2 \leftarrow EFG, \;\; 4 \leftarrow CDF, \;\; 7 \leftarrow BCE, \;\; 9 \leftarrow B, \;\; 11 \leftarrow \langle ABD, A \rangle,$$

node 11 taking two leaves at once, and leaving the path $1 \to 2 \to 4 \to 7 \to 9 \to 11$. *Compress:* node 9 is barred from a chain because its single child 11 is now a leaf, so the unique maximal chain is $1, 2, 4, 7$, and absorbing N_i into N_{i+1} for odd i gives $2 \leftarrow \langle C, EFG \rangle$ and $7 \leftarrow \langle CDF, BCE \rangle$, leaving $2 \to 7 \to 9 \to 11$.
(ii) *Rake* 11 into 9: $9 \leftarrow \big\langle \langle ABD, A \rangle, B \big\rangle$, leaving $2 \to 7 \to 9$. *Compress:* the only maximal chain is the single node 2, and a chain of length one has no odd i with a successor, so nothing moves.
(iii) *Rake* 9 into 7:

$$7 \leftarrow \Big\langle \big\langle \langle ABD, A \rangle, B \big\rangle, \; \langle CDF, BCE \rangle \Big\rangle;$$

no node now has a nonleaf single child, so *compress* does nothing.

(iv) *Rake* 7 into 2:

$$2 \leftarrow \Big\langle \Big\langle \langle\langle ABD, A\rangle, B\rangle, \langle CDF, BCE\rangle \Big\rangle,\ \langle C, EFG\rangle \Big\rangle,$$

and one node remains, so this label is returned. Naming the new internal nodes R (root), X, Z, X_1, X_2, Y, it is the dtree

```
R |- X |  |- X1 |  |  |- Y |  |  |  |- leaf ABD |  |  |  `- leaf A |  |  `- leaf B |  `- X2 |
↪  |- leaf CDF |     `- leaf BCE
`- Z |- leaf C
   `- leaf EFG
```

Its height is 4 against 6 for the dtree of Figure 9.28, the $O(\log n)$ height promised by Theorem 9.15 for $n = 13$. Clusters of the original dtree, leaf cutsets being read as $\text{vars}(L) \setminus \text{acutset}(L)$:

node	vars	cutset	context	cluster
1	$ABCDEFG$	C	$\varnothing$	C
2	$ABCDEFG$	EF	C	CEF
3	C	$\varnothing$	C	C
4	$ABCDEF$	D	CEF	$CDEF$
5	EFG	G	EF	EFG
6	CDF	$\varnothing$	CDF	CDF
7	$ABCDE$	B	CDE	$BCDE$
8	BCE	$\varnothing$	BCE	BCE
9	ABD	$\varnothing$	BD	BD
10	B	$\varnothing$	B	B
11	ABD	A	BD	ABD
12	ABD	$\varnothing$	ABD	ABD
13	A	$\varnothing$	A	A

giving width 3 and largest context $w = 3$; and for the balanced dtree:

node	vars	cutset	context	cluster
R	$ABCDEFG$	CEF	$\varnothing$	CEF
X	$ABCDEF$	BD	CEF	$BCDEF$
X_1	ABD	$\varnothing$	BD	BD
Y	ABD	A	BD	ABD
ABD	ABD	$\varnothing$	ABD	ABD
A	A	$\varnothing$	A	A
B	B	$\varnothing$	B	B
X_2	$BCDEF$	$\varnothing$	$BCDEF$	$BCDEF$
CDF	CDF	$\varnothing$	CDF	CDF
BCE	BCE	$\varnothing$	BCE	BCE
Z	$CEFG$	$\varnothing$	CEF	CEF
C	C	$\varnothing$	C	C
EFG	EFG	G	EF	EFG

Hence cutset width 3, context width 5 and width 4, all inside the Theorem 9.15 guarantees $\le w = 3$, $\le 2w = 6$ and $\le 3w - 1 = 8$: balancing bought two units of height for one of width. COMPOSE connects its arguments arbitrarily, so this is one legitimate output of BAL DT rather than the only one.

Problem 9.25. Define a jointree for an undirected graph G as a pair (T, C), where T is a tree and C is a function that maps each node i in the tree T to a label C_i, called a cluster, that satisfies the following conditions:

- The cluster C_i is a set of nodes in graph G.
- For every edge $X - Y$ in graph G, the variables X and Y appear in some cluster C_i.
- The clusters of tree T satisfy the jointree property: if a node of G appears in two clusters C_i and C_j, it appears in every cluster C_k on the path connecting i and j in T.

Show that every clique of G must be contained in some cluster of the jointree. Show also that if G is the moral graph of some DAG $G^\star$, then (T, C) is also a jointree for DAG $G^\star$. Note: this definition

of a jointree is known as a tree decomposition (for graph G) in the graph-theoretic literature.

Solution. Both claims reduce to the Helly property of subtrees. For a node X of G put $T_X = \{\, i \in T : X \in C_i \,\}$; each T_X is connected, since the jointree property places the whole i-to-j path inside T_X whenever $i, j \in T_X$.
Helly. Nonempty pairwise-intersecting subtrees $S_1, \ldots, S_k$ of T share a node. Root T arbitrarily. Each S_i has a unique node h_i of minimum depth, since two of equal minimum depth would force their least common ancestor, which lies on the connecting path and hence in S_i, to be shallower; and every $v \in S_i$ is a descendant of h_i, the least common ancestor of v and h_i lying in S_i at depth $\le \mathrm{depth}(h_i)$ and so equalling h_i. Pick j maximising $\mathrm{depth}(h_j)$ and fix any i. A node $v \in S_i \cap S_j$ descends from both h_i and h_j, so these are comparable and h_i is an ancestor-or-equal of h_j; then h_j lies on the h_i-to-v path, which lies in the connected S_i. Hence $h_j \in S_1 \cap \cdots \cap S_k$. □
Cliques. Let K be a clique with $|K| \ge 2$. Any two distinct $X, Y \in K$ are adjacent, so by the second condition some C_i contains both, giving $i \in T_X \cap T_Y$; the subtrees T_X, $X \in K$, are therefore nonempty and pairwise intersecting, and Helly supplies a node i with $K \subseteq C_i$. If $K = \{X\}$ and X has a neighbour, the cluster covering that edge serves; if X is isolated the three conditions never mention X, and the claim holds exactly under the usual extra tree-decomposition requirement that every node of G occur in some cluster, which we adopt.
Moral graphs. Take $G = G^\star_m$. Conditions 1 and 3 of Definition 9.13 are verbatim the first and third conditions above, the moral graph having exactly the nodes of $G^\star$ and the jointree property mentioning no graph at all. For condition 2, every family $X\mathbf{U}$ is a clique of G: Definition 9.1 undirects each edge $U \to X$, making X adjacent to every $U \in \mathbf{U}$, and connects each pair $U, U' \in \mathbf{U}$ as parents of the common child X. The previous paragraph then yields a cluster $C_i \supseteq X\mathbf{U}$. ■

Problem 9.26. Define a dtree for an undirected graph G as a pair (T, vars), where T is a full binary tree whose leaves are in one-to-one correspondence with the graph edges and $\mathrm{vars}(\cdot)$ is a function that maps each leaf node L in the dtree to the two variables $\mathrm{vars}(L)$ of the corresponding edge. Define the cutset, context, and cluster as they are defined for dtrees of DAGs, that is,

$$\begin{aligned}
\mathrm{vars}(N) &= \mathrm{vars}(N^l) \cup \mathrm{vars}(N^r) && \text{for nonleaf } N,\\
\mathrm{cutset}(N) &= \big(\mathrm{vars}(N^l) \cap \mathrm{vars}(N^r)\big) \setminus \mathrm{acutset}(N) && \text{for nonleaf } N,\\
\mathrm{acutset}(N) &= \bigcup\nolimits_{N^\star} \mathrm{cutset}(N^\star) && \text{over ancestors } N^\star \text{ of } N,\\
\mathrm{context}(N) &= \mathrm{vars}(N) \cap \mathrm{acutset}(N),
\end{aligned}$$

with $\mathrm{cluster}(N) = \mathrm{vars}(N)$ at a leaf and $\mathrm{cluster}(N) = \mathrm{cutset}(N) \cup \mathrm{context}(N)$ otherwise. Show that the dtree clusters satisfy the jointree property. Show also that every clique of G must be contained in some cluster of the dtree. Note: this definition of a dtree is known as a branch decomposition (for graph G) in the graph-theoretic literature.

Solution. Both claims follow once each variable is pinned to a subtree of clusters. Fix a variable X occurring in some leaf label, let $L_X \neq \varnothing$ be the leaves whose label contains X, and let M_X be their least common ancestor. Unfolding the recursion for vars,

$$X \in \mathrm{vars}(N) \iff \text{some leaf of } L_X \text{ lies at or below } N. \tag{1}$$

Nothing below uses the leaf labels beyond their being sets of variables, so the same argument proves Theorem 9.10 for dtrees of DAGs.
X lies in exactly one cutset. (i) At a proper ancestor N of M_X, all of L_X sits in one child subtree, so by (1) $X \notin \mathrm{vars}(N^l) \cap \mathrm{vars}(N^r)$ and $X \notin \mathrm{cutset}(N)$. (ii) If M_X is a nonleaf, each of its child subtrees holds a leaf of L_X, since otherwise that child would be a deeper common ancestor; so by (1) X lies in both children's vars, while (i) gives $X \notin \mathrm{acutset}(M_X)$, and hence $X \in \mathrm{cutset}(M_X)$. (iii) At a proper descendant N of M_X, (ii) puts X in $\mathrm{acutset}(N)$, which the set difference then removes. (iv) At an N incomparable with M_X, (1) gives $X \notin \mathrm{vars}(N) \supseteq \mathrm{cutset}(N)$. Thus X lies in $\mathrm{cutset}(N)$ exactly when

$N = M_X$ and M_X is a nonleaf, whence

$$X \in \text{acutset}(N) \iff M_X \text{ is a proper ancestor of } N. \tag{2}$$

The clusters holding X.

$$X \in \text{cluster}(N) \iff N \text{ is } M_X \text{ or below it, and } X \in \text{vars}(N). \tag{3}$$

At a leaf cluster = vars, and every leaf containing X belongs to L_X and so lies at or below M_X. At a nonleaf N: if $N = M_X$ then $X \in \text{cutset}(N)$ by (ii); if N is a proper descendant with $X \in \text{vars}(N)$ then (2) puts X in context(N). Conversely $X \in \text{cluster}(N)$ forces $X \in \text{vars}(N)$, since cutset and context both sit inside vars(N), so N is comparable with M_X by (1); and a proper ancestor is excluded by (i) and (2) together.

Jointree property. By (3), $S_X = \{N : X \in \text{cluster}(N)\}$ is the set of nodes at or below M_X having a leaf of L_X at or below them, and each such N is joined to M_X inside S_X, every node on the path from N up to M_X still having that leaf below it. So S_X is connected, hence contains the path between any two of its nodes, which is the jointree property. Variables occurring in no leaf label occur in no cluster, and the property is vacuous for them.

Cliques. The pair $(T, \text{cluster})$ meets the three conditions of Exercise 9.25: clusters are sets of nodes of G; each edge $X - Y$ is the label of a leaf L with $\text{cluster}(L) = \text{vars}(L) = \{X, Y\}$; and the jointree property holds by the above. Exercise 9.25 then places every clique of G inside some cluster, with the same degenerate isolated-node case. ■

Problem 9.27. Let G be a DAG, G_m its moral graph, and consider a jointree (T, C) for DAG G with width w. Let G_d be the graph that results from connecting in G_m every pair of variables that appear together in some jointree cluster. Show that G_d is triangulated and that its treewidth is w.

Solution. Peeling leaves off the jointree, as Algorithm 21 (JT2EO) does, produces a perfect elimination order for G_d, so G_d is triangulated, and its treewidth is w. Throughout, $|C_i| \leq w + 1$ for every cluster with equality for at least one, the jointree having width w.

Two facts. (F1) Every cluster is a clique of G_d, since G_d joins every pair occurring together in a cluster. (F2) Every edge of G_d lies inside some cluster: the added edges do by definition, and an edge of G_m is by Definition 9.1 either an undirected $U \to X$, inside the family of X, or a link between two parents of a common child Z, inside the family of Z – and condition 2 of Definition 9.13 puts every family inside a cluster. In particular every variable occurs in a cluster, lying in its own family.

The order. While the current tree T' has more than one node, pick a leaf i with neighbour j, append $C_i \setminus C_j$ to π in any order and delete i; at the end append the variables of the one remaining cluster. Only leaves are deleted, so T' is always a connected subtree of T and each deleted node has a unique *attachment point* in T', the first T'-node on its path in. Write B_i for $\{i\}$ together with the deleted nodes attached at i, so that T is the disjoint union of the B_i, $i \in T'$.

(i) *For a leaf i with neighbour j and $X \in C_i \setminus C_j$, every cluster containing X is C_k with $k \in B_i$.* Were $k \in B_m$ with $m \in T'$, $m \neq i$, the k-to-i path would run to m and then along T', ending with the edge $j - i$ as i is a leaf, so the jointree property would force $X \in C_j$. Since $B_i \cap T' = \{i\}$, C_i is the only surviving cluster containing X. Hence peeling i deletes exactly $C_i \setminus C_j$ from $\bigcup_{k \in T'} C_k$, the surviving variables V are always $\bigcup_{k \in T'} C_k$, and each variable is eliminated exactly once.

(ii) *Each $X \in C_i \setminus C_j$ is simplicial when eliminated, of degree $\leq |C_i| - 1 \leq w$.* Inductively no fill-in edge has been added, so the current graph is G_d restricted to V. A surviving G_d-neighbour Y of X shares a cluster C_k with X by (F2), and $k \in B_i$ by (i); if $k = i$ then $Y \in C_i$, and otherwise $Y \in C_m$ for some $m \in T'$ and the k-to-m path crosses i, so the jointree property again gives $Y \in C_i$. All surviving neighbours of X thus lie in $C_i \setminus \{X\}$ and are pairwise adjacent by (F1). At the last stage $V = C_i$ is a clique by (F1), so its vertices are simplicial in turn, again of degree $\leq w$.

No fill-in edge is ever added, so $(G_d)_\pi = G_d$, π is a perfect elimination order in the sense of Definition 9.15, and G_d is triangulated by Theorem 9.17. By (ii) every cluster induced by π has at most $w + 1$ variables, so $\text{treewidth}(G_d) \leq \text{width}(\pi, G_d) \leq w$; and some C_i has $w + 1$ variables and is a clique of G_d

by (F1), so Theorem 9.1 gives $\text{treewidth}(G_d) \geq w$. Hence

$$\text{treewidth}(G_d) \;=\; w. \qquad \blacksquare$$

Problem 9.28. Show that the treewidth of a triangulated graph equals the size of its maximal clique minus one; that is, if G is triangulated and $\omega(G)$ denotes the number of nodes in a largest clique of G, then $\text{treewidth}(G) = \omega(G) - 1$.

Solution. $\text{treewidth}(G) = \omega(G) - 1$, where $\omega(G)$ is the size of a largest clique; the statement's "maximal clique" must be read that way, a triangulated graph generally having inclusion-maximal cliques of several sizes.
Lower bound. G has a clique of size $\omega(G)$, so Theorem 9.1 gives $\text{treewidth}(G) \geq \omega(G) - 1$, triangulation unused.
Upper bound. By Theorem 9.17 there is a perfect elimination order π, so $G_\pi = G$ by Definition 9.15. Let $G_1 = G$ and let G_{i+1} come from G_i by pairwise connecting the neighbours of $\pi(i)$ and deleting $\pi(i)$. Elimination only ever adds edges and $G_\pi = G$, so every edge added at every step was already present in G; hence no step alters the edges among survivors, making G_i the subgraph of G induced by $\pi(i), \ldots, \pi(n)$, and the neighbours of $\pi(i)$ in G_i pairwise adjacent in G. The induced cluster

$$C_i \;=\; \{\pi(i)\} \cup \{\text{neighbours of } \pi(i) \text{ in } G_i\}$$

is therefore a clique of G, those neighbours being adjacent to $\pi(i)$ in G as well, so $|C_i| \leq \omega(G)$ for every i and

$$\text{treewidth}(G) \leq \text{width}(\pi, G) = \max_i |C_i| - 1 \leq \omega(G) - 1. \qquad \blacksquare$$

9 Most Likely Instantiations

Contents

9.1 Exercises 10.1–10.7 140
9.2 Exercises 10.8–10.14 147
9.3 Exercises 10.15–10.21 153
9.4 Exercises 10.22–10.22 159

9.1 Exercises 10.1–10.7

Problem 10.1. Compute the MPE probability and a corresponding MPE instantiation for the Bayesian network in Figure 10.1, given evidence $A = \mathsf{no}$ and using Algorithm 27, $\mathrm{VE}_{\mathrm{MPE}}$.
Figure 10.1 is the following network. Variable S is the gender of an individual, C is a condition that is more likely in males, T_1 and T_2 are two tests for the condition, and A records whether the two tests agree on the individual. The edges are

$$S \to C, \quad S \to T_2, \quad C \to T_1, \quad C \to T_2, \quad T_1 \to A, \quad T_2 \to A.$$

The CPTs are:

S	θ_s
male	.55
female	.45

S	C	$\theta_{c\|s}$
male	yes	.05
male	no	.95
female	yes	.01
female	no	.99

C	T_1	$\theta_{t_1\|c}$
yes	+ve	.80
yes	−ve	.20
no	+ve	.20
no	−ve	.80

S	C	T_2	$\theta_{t_2\|c,s}$
male	yes	+ve	.80
male	yes	−ve	.20
male	no	+ve	.20
male	no	−ve	.80
female	yes	+ve	.95
female	yes	−ve	.05
female	no	+ve	.05
female	no	−ve	.95

T_1	T_2	A	$\theta_{a\|t_1,t_2}$
+ve	+ve	yes	1
+ve	+ve	no	0
+ve	−ve	yes	0
+ve	−ve	no	1
−ve	+ve	yes	0
−ve	+ve	no	1
−ve	−ve	yes	1
−ve	−ve	no	0

Solution. $MPE_{Pr}(A = \mathsf{no}) = .084645$, attained at the unique instantiation $S = \text{female}$, $C = \mathsf{no}$, $T_1 = +\text{ve}$, $T_2 = -\text{ve}$, $A = \mathsf{no}$. Write $e : A = \mathsf{no}$.
Line 1 of $\mathrm{VE}_{\mathrm{MPE}}$ calls pruneEdges(N, e), which removes no edge since the sole evidence variable A is a leaf; only $\theta_{a|t_1,t_2}$ is reduced, to the extended factor

$$f^e_A(T_1, T_2, A) = \begin{cases} \theta_{a|t_1,t_2}, & a = \mathsf{no} \\ 0, & a = \mathsf{yes}, \end{cases}$$

every row of every factor carrying the trivial instantiation $\top$ as Definition 10.2 requires. Take $\pi = A, T_1, T_2, C, S$, of width 2: the eliminations work on $\{T_1, T_2, A\}$, $\{C, T_1, T_2\}$, $\{S, C, T_2\}$, $\{S, C\}$, $\{S\}$. Maximizing A out of the only factor mentioning it, f^e_A, records $A = \mathsf{no}$ throughout, that being the

only value with nonzero entries:

T_1	T_2	g_1	$g_1[\cdot]$
+ve	+ve	0	$A = \mathsf{no}$
+ve	−ve	1	$A = \mathsf{no}$
−ve	+ve	1	$A = \mathsf{no}$
−ve	−ve	0	$A = \mathsf{no}$

So g_1 is the indicator of "the two tests disagree". Eliminating T_1, the factors mentioning it are $\theta_{T_1|C}$ and g_1, with product

C	T_1	T_2	product
yes	+ve	+ve	0
yes	+ve	−ve	.80
yes	−ve	+ve	.20
yes	−ve	−ve	0
no	+ve	+ve	0
no	+ve	−ve	.20
no	−ve	+ve	.80
no	−ve	−ve	0

Maximizing out T_1 gives

C	T_2	g_2	$g_2[\cdot]$
yes	+ve	.20	$T_1 = -\text{ve}$, $A = \mathsf{no}$
yes	−ve	.80	$T_1 = +\text{ve}$, $A = \mathsf{no}$
no	+ve	.80	$T_1 = -\text{ve}$, $A = \mathsf{no}$
no	−ve	.20	$T_1 = +\text{ve}$, $A = \mathsf{no}$

the instantiations recorded by g_1 being carried along by the multiplication rule of Definition 10.2. Eliminating T_2, the factors $\theta_{T_2|C,S}$ and g_2 have product

S	C	T_2	product
male	yes	+ve	$.80 \times .20 = .16$
male	yes	−ve	$.20 \times .80 = .16$
male	no	+ve	$.20 \times .80 = .16$
male	no	−ve	$.80 \times .20 = .16$
female	yes	+ve	$.95 \times .20 = .19$
female	yes	−ve	$.05 \times .80 = .04$
female	no	+ve	$.05 \times .80 = .04$
female	no	−ve	$.95 \times .20 = .19$

Maximizing out T_2 gives (ties for $S =$ male are broken arbitrarily, which Definition 10.2 permits; here we keep $T_2 = +\text{ve}$):

S	C	g_3	$g_3[\cdot]$
male	yes	.16	$T_2 = +\text{ve}$, $T_1 = -\text{ve}$, $A = \mathsf{no}$
male	no	.16	$T_2 = +\text{ve}$, $T_1 = -\text{ve}$, $A = \mathsf{no}$
female	yes	.19	$T_2 = +\text{ve}$, $T_1 = -\text{ve}$, $A = \mathsf{no}$
female	no	.19	$T_2 = -\text{ve}$, $T_1 = +\text{ve}$, $A = \mathsf{no}$

Eliminating C, the factors $\theta_{C|S}$ and g_3 have product

S	C	product
male	yes	$.05 \times .16 = .0080$
male	no	$.95 \times .16 = .1520$
female	yes	$.01 \times .19 = .0019$
female	no	$.99 \times .19 = .1881$

Maximizing out C:

S	g_4	$g_4[\cdot]$
male	.1520	$C = \mathsf{no}$, $T_2 = +\text{ve}$, $T_1 = -\text{ve}$, $A = \mathsf{no}$
female	.1881	$C = \mathsf{no}$, $T_2 = -\text{ve}$, $T_1 = +\text{ve}$, $A = \mathsf{no}$

Finally θ_S and g_4 give

$$\text{male} : .55 \times .1520 = .083600,$$
$$\text{female} : .45 \times .1881 = .084645.$$

so maximizing out S yields the trivial factor

$$f(\top) = .084645,$$
$$f[\top] : S = \text{female},\ C = \mathsf{no},\ T_1 = +\text{ve},$$
$$T_2 = -\text{ve},\ A = \mathsf{no}.$$

Uniqueness is visible in the joint of Table 10.1, whose rows compatible with $A = \mathsf{no}$ are

S	C	T_1	T_2	$Pr(\cdot)$
male	yes	+ve	−ve	.004400
male	yes	−ve	+ve	.004400
male	no	+ve	−ve	.083600
male	no	−ve	+ve	.083600
female	yes	+ve	−ve	.000180
female	yes	−ve	+ve	.000855
female	no	+ve	−ve	.084645
female	no	−ve	+ve	.017820

whose strict maximum is the recovered row, .084645.

Problem 10.2. Consider the Bayesian network in Figure 10.1. What is the most likely outcome of the two tests T_1 and T_2 for a female on whom the tests came out different? Use Algorithm 30, VE_{MAP}, to answer this question.
Figure 10.1 has nodes S (gender), C (a condition), T_1, T_2 (two tests for the condition), and A (whether the two tests agree), with edges

$$S \to C, \quad S \to T_2, \quad C \to T_1, \quad C \to T_2, \quad T_1 \to A, \quad T_2 \to A.$$

Its CPTs are

S	θ_s
male	.55
female	.45

S	C	$\theta_{c\|s}$	C	T_1	$\theta_{t_1\|c}$
male	yes	.05	yes	+ve	.80
male	no	.95	yes	−ve	.20
female	yes	.01	no	+ve	.20
female	no	.99	no	−ve	.80

S	C	T_2	$\theta_{t_2\|c,s}$
male	yes	+ve	.80
male	yes	−ve	.20
male	no	+ve	.20
male	no	−ve	.80
female	yes	+ve	.95
female	yes	−ve	.05
female	no	+ve	.05
female	no	−ve	.95

and $\theta_{a|t_1,t_2} = 1$ for $A = \mathsf{yes}$ exactly when $T_1 = T_2$, and $\theta_{a|t_1,t_2} = 1$ for $A = \mathsf{no}$ exactly when $T_1 \neq T_2$ (so A is a deterministic agreement indicator).

Solution. The most likely outcome is $T_1 = +\text{ve}$, $T_2 = -\text{ve}$, with $MAP_{Pr}(\mathbf{M}, e) = .084825$. This is the MAP query with $\mathbf{M} = \{T_1, T_2\}$ and evidence

$$e : S = \text{female},\ A = \mathsf{no},$$

A being the deterministic agreement indicator; $\mathbf{E} \cap \mathbf{M} = \emptyset$, as $\mathrm{VE}_{\mathrm{MAP}}$ requires. Line 1 calls pruneNetwork$(N, \mathbf{M}, e)$: node pruning removes nothing, the only leaf A being evidence, and edge pruning deletes $S \to C$ and $S \to T_2$ out of the evidence node S. Line 3 reduces every remaining CPT by e, giving

$$\begin{aligned} f_S(S) &= \theta^e_S : \ f_S(\text{female}) = .45, \ f_S(\text{male}) = 0, \\ f_C(C) &= \theta_{C|S=\text{female}} : \ \theta_{\mathsf{yes}} = .01, \ \theta_{\mathsf{no}} = .99, \\ f_1(C, T_1) &= \theta_{T_1|C}, \\ f_2(C, T_2) &= \theta_{T_2|C,S=\text{female}}, \\ f_A(T_1, T_2, A) &= \theta^e_{A|T_1,T_2}, \end{aligned}$$

the last being zero on every row with $A = \mathsf{yes}$. Definition 10.4 forces the MAP variables last, so take the $\mathbf{M}$-constrained order

$$\pi = S, \ A, \ C, \ T_1, \ T_2,$$

of width 2, summing out S, A, C and then maximizing out T_1, T_2. Now f_S is nonzero only at $S = \text{female}$, so $\sum_S f_S = .45$, and f_A is nonzero only at $A = \mathsf{no}$, so summing out A leaves the disagreement indicator

T_1	T_2	$g(T_1, T_2)$
+ve	+ve	0
+ve	−ve	1
−ve	+ve	1
−ve	−ve	0

The factors mentioning C are f_C, f_1 and f_2, with product

C	T_1	T_2	$f_C f_1 f_2$
yes	+ve	+ve	$.01 \times .80 \times .95 = .00760$
yes	+ve	−ve	$.01 \times .80 \times .05 = .00040$
yes	−ve	+ve	$.01 \times .20 \times .95 = .00190$
yes	−ve	−ve	$.01 \times .20 \times .05 = .00010$
no	+ve	+ve	$.99 \times .20 \times .05 = .00990$
no	+ve	−ve	$.99 \times .20 \times .95 = .18810$
no	−ve	+ve	$.99 \times .80 \times .05 = .03960$
no	−ve	−ve	$.99 \times .80 \times .95 = .75240$

Summing over C gives

T_1	T_2	$h(T_1, T_2)$
+ve	+ve	.01750
+ve	−ve	.18850
−ve	+ve	.04150
−ve	−ve	.75250

At this point the remaining factors are the constant .45, the indicator g, and h; their product is exactly the joint marginal $Pr(T_1, T_2, e)$:

T_1	T_2	$Pr(T_1, T_2, S = \text{female}, A = \mathsf{no})$
+ve	+ve	0
+ve	−ve	$.45 \times .18850 = .084825$
−ve	+ve	$.45 \times .04150 = .018675$
−ve	−ve	0

Maximizing out T_1 with the extended factors of Section 10.2.1,

T_2	$\max_{T_1}$	recorded
+ve	.018675	$T_1 = -\text{ve}$
−ve	.084825	$T_1 = +\text{ve}$

and then $\max_{T_2}$ of these two numbers is .084825, recorded with $T_2 = -\text{ve}$, so the returned trivial factor is

$$f(\top) = .084825, \qquad f[\top] : T_1 = +\text{ve}, \ T_2 = -\text{ve}.$$

that is, $MAP_{Pr}(\{T_1, T_2\}, e) = Pr(T_1 = +\text{ve}, T_2 = -\text{ve}, e) = .084825$.

Problem 10.3. Show that the MAP example in Section 10.3.1 admits another MAP instantiation with the same probability .242720.
That example uses the digital-circuit network of Figure 10.2, with MAP variables $\mathbf{M} = \{I, J\}$ and evidence $e : O = \text{true}$. All five variables are binary with values true/false, and the edges are

$$J \to Y, \quad I \to X, \quad J \to X, \quad X \to O, \quad Y \to O.$$

The CPTs are

I	θ_i	J	θ_j
true	.5	true	.5
false	.5	false	.5

J	Y	$\theta_{y\|j}$
true	true	.01
true	false	.99
false	true	.99
false	false	.01

I	J	X	$\theta_{x\|i,j}$
true	true	true	.95
true	true	false	.05
true	false	true	.05
true	false	false	.95
false	true	true	.05
false	true	false	.95
false	false	true	.05
false	false	false	.95

X	Y	O	$\theta_{o\|x,y}$
true	true	true	.98
true	true	false	.02
true	false	true	.98
true	false	false	.02
false	true	true	.98
false	true	false	.02
false	false	true	.02
false	false	false	.98

Running VE_{MAP} with the order $\pi = O, Y, X, I, J$ produces, after summing out O, Y and X, the factor

I	J	f_1
true	true	.93248
true	false	.97088
false	true	.07712
false	false	.97088

together with the priors $f_2(I)$ and $f_3(J)$, and the algorithm reports the MAP instantiation $I =$ true, $J =$ false with MAP probability .242720.

Solution. $I =$ false, $J =$ false is a second MAP instantiation of probability .242720. VE_{MAP} with the order O, Y, X, I, J leaves $f_1(I, J)$ together with the uniform priors $f_2(I) \equiv f_3(J) \equiv .5$, so that

$$Pr(I, J, O = \text{true}) \;=\; f_1(I, J)\, f_2(I)\, f_3(J) \;=\; .25 \cdot f_1(I, J).$$

Reading off all four entries:

I	J	$Pr(I, J, O = \mathsf{true})$
true	true	$.25 \times .93248 = .233120$
true	false	$.25 \times .97088 = .242720$
false	true	$.25 \times .07712 = .019280$
false	false	$.25 \times .97088 = .242720$

so the maximum .242720 is attained twice and

$$MAP(\{I, J\}, O = \mathsf{true}) \;=\; \{\,(I = \mathsf{true}, J = \mathsf{false}),\; (I = \mathsf{false}, J = \mathsf{false})\,\}.$$

The tie is structural. The CPT of O is the noisy or $Pr(O = \mathsf{true} \mid x, y) = .98 - .96 \cdot [\,x = \mathsf{false}$ and $y = \mathsf{false}\,]$, and $X \perp\!\!\!\perp Y \mid I, J$ in this network, so

$$f_1(i, j) \;=\; Pr(O = \mathsf{true} \mid i, j) \;=\; .98 - .96 \cdot Pr(x_f \mid i, j)\, Pr(y_f \mid j),$$

writing x_f for $X = \mathsf{false}$ and y_f for $Y = \mathsf{false}$. At $J = \mathsf{false}$ the CPT of X has $\theta_{X=\mathsf{true}|i,\mathsf{false}} = .05$ for both values of i, so $Pr(x_f \mid i, \mathsf{false}) = .95$ whatever i is, and Y never depended on I; hence $f_1(i, \mathsf{false}) = .98 - .96 \times .95 \times .01 = .97088$ for both i, and the uniform prior on I carries the tie through. In the book's trace it surfaces as $\big(\max_I f_1 f_2\big)(J = \mathsf{false}) = \max(.485440, .485440)$, where Definition 10.2 permits recording either maximizer; recording $I = \mathsf{false}$ returns the second instantiation.

Problem 10.4. Construct a factor f over variables X and Y such that

$$\sum_Y \max_X f \;\neq\; \max_X \sum_Y f.$$

Solution. Take X and Y binary with values x_1, x_2 and y_1, y_2, and let f be the indicator of "X and Y take matching indices":

X	Y	f
x_1	y_1	1
x_1	y_2	0
x_2	y_1	0
x_2	y_2	1

Maximizing out X first (Definition 10.1) gives $\big(\max_X f\big)(y_1) = \big(\max_X f\big)(y_2) = 1$, hence $\sum_Y \max_X f = 2$; summing out Y first gives $\big(\sum_Y f\big)(x_1) = \big(\sum_Y f\big)(x_2) = 1$, hence $\max_X \sum_Y f = 1$. No single x maximizes $f(\cdot, y)$ at both values of y, which is exactly what defeats equality in Theorem 10.4.

Problem 10.5. Construct a factor f over variables X and Y such that

$$\sum_Y \max_X f \;=\; \max_X \sum_Y f.$$

Solution. Take X and Y binary with values x_1, x_2 and y_1, y_2, and let

X	Y	f
x_1	y_1	2
x_1	y_2	3
x_2	y_1	1
x_2	y_2	0

whose row x_1 dominates row x_2 pointwise, so that the single value x_1 maximizes $f(\cdot, y)$ at every y. Then $\max_X f$ takes the values $\max(2, 1) = 2$ at y_1 and $\max(3, 0) = 3$ at y_2, giving $\sum_Y \max_X f = 5$; and $\sum_Y f$ takes the values $2 + 3 = 5$ at x_1 and $1 + 0 = 1$ at x_2, giving $\max_X \sum_Y f = 5$ as well.

Problem 10.6. Prove Theorem 10.1: if f_1 and f_2 are factors and if variable X appears only in f_2, then

$$\max_X f_1 f_2 \;=\; f_1 \max_X f_2.$$

Here $\max_X f$ is the factor of Definition 10.1: if f is a factor over variables $\mathbf{X}$ and $X \in \mathbf{X}$, then $\max_X f$ is the factor over $\mathbf{Y} = \mathbf{X} \setminus \{X\}$ defined by $\big(\max_X f\big)(\mathbf{y}) = \max_x f(x, \mathbf{y})$.

Solution. Both sides are factors over $\mathbf{W} = (\mathbf{Y} \cup \mathbf{Z}) \setminus \{X\}$, where f_1 is over $\mathbf{Y}$ and f_2 over $\mathbf{Z}$ with $X \in \mathbf{Z}$, $X \notin \mathbf{Y}$: on the right $\max_X f_2$ is a factor over $\mathbf{Z} \setminus \{X\}$, and $\mathbf{Y} \cup (\mathbf{Z} \setminus \{X\}) = \mathbf{W}$ because $X \notin \mathbf{Y}$. Fix an instantiation $\mathbf{w}$ of $\mathbf{W}$ and let $\mathbf{y}$, $\mathbf{z}'$ be its restrictions to $\mathbf{Y}$ and $\mathbf{Z} \setminus \{X\}$; the instantiation $\mathbf{y}$ is free of X precisely because X appears only in f_2, so $(f_1 f_2)(x\mathbf{w}) = f_1(\mathbf{y})\, f_2(x\mathbf{z}')$ for each value x, and Definition 10.1 gives

$$\begin{aligned}\Big(\max_X f_1 f_2\Big)(\mathbf{w}) &= \max_x\; f_1(\mathbf{y})\, f_2(x\mathbf{z}') \\ &= f_1(\mathbf{y}) \cdot \max_x f_2(x\mathbf{z}') \\ &= \Big(f_1 \max_X f_2\Big)(\mathbf{w}).\end{aligned}$$

The middle step pulls out $f_1(\mathbf{y})$, which is constant in x and nonnegative by Definition 6.1: for $c \geq 0$ one has $\max_x c\, v_x = c \max_x v_x$, both sides vanishing when $c = 0$. Since $\mathbf{w}$ was arbitrary, the factors are equal. ■

Problem 10.7. Consider a naive Bayes structure with edges $C \to A_1, \ldots, C \to A_n$. What is the complexity of computing the MPE probability for this network using Algorithm 27, $\mathrm{VE}_{\mathrm{MPE}}$? What is the complexity of computing the MAP probability for MAP variables $A_1, \ldots, A_n$ using Algorithm 30, $\mathrm{VE}_{\mathrm{MAP}}$? How does the complexity of these computations change when we have evidence on variable C?

Solution. MPE costs $O(n \exp(1))$, MAP with $\mathbf{M} = \{A_1, \ldots, A_n\}$ costs $O(n \exp(n))$, and evidence on C drops both to $O(n \exp(0))$. Here d bounds the cardinality of a variable, and elimination of width w on N variables costs $O(N \exp(w))$ time and space.
MPE. Under $\pi = A_1, \ldots, A_n, C$, each A_i occurs in the single CPT $\theta_{A_i|C}$, so Theorem 10.1 lets it be maximized out on its own, leaving $\lambda_i(C) = \max_{a_i} \theta_{a_i|C}$; eliminating C then multiplies $\theta_C, \lambda_1, \ldots, \lambda_n$ and maximizes:

$$MPE_{Pr} \;=\; \max_c\; \theta_c \prod_{i=1}^{n} \max_{a_i} \theta_{a_i|c}.$$

Every cluster has two variables, so $w = 1$ and the cost is $O\big((n+1)\exp(1)\big) = O(n)$, or $O(nd^2)$ counting table entries.
MAP. Definition 10.4 puts the MAP variables last, so C, the only non-MAP variable, is eliminated first; nothing prunes away beforehand, pruneNetwork removing only leaves outside $\mathbf{M} \cup \mathbf{E}$ and every leaf A_i being a MAP variable. That step multiplies all $n+1$ CPTs into a factor over $\{C, A_1, \ldots, A_n\}$ and sums C out to

$$h(A_1, \ldots, A_n) \;=\; Pr(A_1, \ldots, A_n) \;=\; \sum_c \theta_c \prod_{i=1}^{n} \theta_{a_i|c},$$

out of which the A_i are then maximized one at a time. The width is n, and as every admissible order has this shape the $\mathbf{M}$-constrained treewidth is exactly n against an unconstrained treewidth of 1; so $\mathrm{VE}_{\mathrm{MAP}}$ costs $O\big((n+1)\exp(n)\big) = O(n\, d^n)$.
Evidence $e : C = c$. Pruning deletes all n edges out of C and reduces the CPTs, leaving an isolated C with θ^e_C and isolated roots A_i with $\lambda_i(A_i) = \theta_{A_i|c}$. No factor now mentions two variables, so every

order, constrained or not, has width 0, and both algorithms return

$$\theta_c \prod_{i=1}^{n} \max_{a_i} \theta_{a_i|c},$$

with $a_i^\star = \arg\max_{a_i} \theta_{a_i|c}$ for each i, together with $C = c$ for MPE. MPE stays linear, its constant improving to $O(nd)$, while MAP falls from exponential to $O(n)$, the evidence d-separating the attributes from one another. In summary:

query	evidence	constrained order needed	width	complexity
MPE	none	no	1	$O(n \exp(1))$
MAP	none	yes	n	$O(n \exp(n))$
MPE	$C = c$	no	0	$O(n \exp(0))$
MAP	$C = c$	yes	0	$O(n \exp(0))$

9.2 Exercises 10.8–10.14

Problem 10.8. True or false: We can compute a MAP instantiation $\text{MAP}(\mathbf{M}, \mathbf{e})$ by computing the MAP instantiations $\mathbf{m}_x = \text{MAP}(\mathbf{M}, \mathbf{e}x)$ for every value x of some variable $X \notin \mathbf{M}$ and then returning $\text{argmax}_{\mathbf{m}_x} Pr(\mathbf{m}_x, \mathbf{e})$. What if $X \in \mathbf{M}$? For each case, either prove or provide a counterexample.

Solution. False when $X \notin \mathbf{M}$, true when $X \in \mathbf{M}$.
(i) $X \notin \mathbf{M}$. The scheme searches only the candidate set $\{\mathbf{m}_x : x\}$, which has at most as many elements as X has values, whereas the true MAP maximizes $Pr(\mathbf{m}, \mathbf{e}) = \sum_x Pr(\mathbf{m}, \mathbf{e}, x)$, and the maximizer of a sum need maximize no single summand — the failure of maximization to commute with summation recorded in Theorem 10.4. For a counterexample let X be a root with values x_1, x_2, let A be its only child with values a_1, a_2, a_3, take $\mathbf{M} = \{A\}$ and $\mathbf{e}$ empty, and use the CPTs

X	Θ_X
x_1	.5
x_2	.5

X	A	$\Theta_{A\|X}$
x_1	a_1	.6
x_1	a_2	0
x_1	a_3	.4
x_2	a_1	0
x_2	a_2	.6
x_2	a_3	.4

whose joint is

A	$Pr(A, x_1)$	$Pr(A, x_2)$	$Pr(A)$
a_1	.3	0	.3
a_2	0	.3	.3
a_3	.2	.2	.4

Then $\mathbf{m}_{x_1} = a_1$ and $\mathbf{m}_{x_2} = a_2$, both of probability .3, while $\text{MAP}(\mathbf{M}, \mathbf{e}) = a_3$ has probability .4; being second best under each value of X and best overall, a_3 is never generated as a candidate.
(ii) $X \in \mathbf{M}$. Write $\mathbf{m} \sim x$ when $\mathbf{m}$ assigns $X = x$. Since $\mathbf{e}x$ contains $X = x$, $Pr(\mathbf{m}, \mathbf{e}, x)$ equals $Pr(\mathbf{m}, \mathbf{e})$ for $\mathbf{m} \sim x$ and 0 otherwise, so $\mathbf{m}_x = \text{argmax}_{\mathbf{m} \sim x} Pr(\mathbf{m}, \mathbf{e})$, ties broken toward an $\mathbf{m}$ compatible with x (an incompatible one can be the argmax only when the whole column is 0, and the final comparison by $Pr(\mathbf{m}_x, \mathbf{e})$ then still returns a correct answer). The instantiations of $\mathbf{M}$ are partitioned by the value they assign X, so the maximization splits blockwise:

$$\begin{aligned} \max_{\mathbf{m}} Pr(\mathbf{m}, \mathbf{e}) &= \max_x \max_{\mathbf{m} \sim x} Pr(\mathbf{m}, \mathbf{e}) \\ &= \max_x Pr(\mathbf{m}_x, \mathbf{e}), \end{aligned}$$

which $\text{argmax}_{\mathbf{m}_x} Pr(\mathbf{m}_x, \mathbf{e})$ attains. ■

Problem 10.9. Suppose that we have a jointree with n clusters and width w. Show that MAP can be solved in $O(n \exp(w))$ time and space if all MAP variables are contained in some jointree cluster.

Solution. Compute the cluster marginal at a cluster r with $\mathbf{M} \subseteq \mathbf{C}_r$, project, and scan. Run the jointree algorithm of Chapter 7 with evidence $\mathbf{e}$, rooted at r and performing only the inward pass, so that cluster i sends its neighbour j on the path to r

$$M_{ij} \;=\; \sum_{\mathbf{C}_i \setminus \mathbf{S}_{ij}} \Big(\prod_{f \in \Phi_i} f^{\mathbf{e}} \Big) \prod_{k \neq j} M_{ki},$$

with Φ_i the CPTs assigned to i and $\mathbf{S}_{ij}$ the separator, leaving at the root

$$Pr(\mathbf{C}_r, \mathbf{e}) \;=\; \Big(\prod_{f \in \Phi_r} f^{\mathbf{e}} \Big) \prod_{k} M_{kr}.$$

Each message is a factor over a separator $\mathbf{S}_{ij} \subseteq \mathbf{C}_i$, hence has at most $\exp(w)$ entries and costs $O(\exp(w))$ after the standard amortization of the message products; there are $n-1$ of them, for $O(n \exp(w))$ time and space. Since $\mathbf{M} \subseteq \mathbf{C}_r$,

$$Pr(\mathbf{M}, \mathbf{e}) \;=\; \sum_{\mathbf{C}_r \setminus \mathbf{M}} Pr(\mathbf{C}_r, \mathbf{e})$$

is a single sum-out over at most $\exp(w)$ rows, and $\operatorname{argmax}_{\mathbf{m}} Pr(\mathbf{m}, \mathbf{e})$ with its value is one further scan of at most $\exp(w)$ rows, both $O(\exp(w))$. Adding the three gives $O(n \exp(w))$ time and space. All summation is completed before any maximization, so Theorem 10.4 poses no obstacle, and the inward pass is exact. ■

Problem 10.10. Show that Algorithm 27, VE MPE, is correct when run with extended factors. In particular, prove the following invariant for this algorithm. Let $\mathbf{X}$ be the variables appearing in the set S maintained by VE MPE, $\mathbf{Y}$ be all other (eliminated) variables, and f be the product of factors in S. Then $f(\mathbf{x}) = Pr(\mathbf{x}, f[\mathbf{x}], \mathbf{e})$ and $f(\mathbf{x}) = \max_{\mathbf{y}} Pr(\mathbf{x}, \mathbf{y}, \mathbf{e})$. Show that this invariant holds before the algorithm reaches Line 5 and remains true after each iteration of Lines 6–8.
For reference, Algorithm 27 is:
VE MPE($\mathcal{N}$, $\mathbf{e}$)
1. $\mathcal{N}' \leftarrow$ pruneEdges($\mathcal{N}, \mathbf{e}$)
2. $\mathbf{Q} \leftarrow$ variables in network $\mathcal{N}'$
3. $\pi \leftarrow$ elimination order of variables $\mathbf{Q}$
4. $S \leftarrow \{f^{\mathbf{e}} : f$ is a CPT of network $\mathcal{N}'\}$
5. for $i = 1$ to $|\mathbf{Q}|$ do
6. $\quad f \leftarrow \prod_k f_k$, where f_k belongs to S and mentions variable $\pi(i)$
7. $\quad f_i \leftarrow \max_{\pi(i)} f$
8. $\quad$ replace all factors f_k in S by factor f_i
9. end for
10. return trivial factor $\prod_{f \in S} f$

Extended factors and their operations are as in Definition 10.2: $\max_X f[\mathbf{y}] = x^\star f[x^\star, \mathbf{y}]$ where $x^\star = \operatorname{argmax}_x f(x, \mathbf{y})$, and $(f_1 f_2)[\mathbf{z}] = f_1[\mathbf{x}] f_2[\mathbf{y}]$ for $\mathbf{x} \sim \mathbf{z}$ and $\mathbf{y} \sim \mathbf{z}$. The algorithm starts with $f[\cdot] = \top$ for every row of every CPT.

Solution. The invariant holds before Line 5 and survives each iteration, so the returned trivial factor carries $f(\top) = \max_{\mathbf{q}} Pr(\mathbf{q}, \mathbf{e})$ with witness $f[\top]$. Let $\mathbf{Q}$ be all network variables, $\mathbf{X}$ those mentioned by S, $\mathbf{Y} = \mathbf{Q} \setminus \mathbf{X}$, and $f = \prod_{g \in S} g$; the invariant is
(I1) $f[\mathbf{x}]$ instantiates exactly $\mathbf{Y}$, with $f(\mathbf{x}) = Pr(\mathbf{x}, f[\mathbf{x}], \mathbf{e})$;
(I2) $f(\mathbf{x}) = \max_{\mathbf{y}} Pr(\mathbf{x}, \mathbf{y}, \mathbf{e})$.
Each variable is recorded into exactly one factor, the f_i built on Line 7 of the iteration where it is

the pivot, and is never mentioned afterwards, so the instantiations attached to distinct factors of S mention disjoint variables and the product rule of Definition 10.2 is well defined.
Before Line 5. Here $\mathbf{X} = \mathbf{Q}$, $\mathbf{Y} = \emptyset$, and every row carries $\top$, an instantiation of $\emptyset$. Reduction gives $h^{\mathbf{e}}(\mathbf{x}) = h(\mathbf{x})$ for $\mathbf{x} \sim \mathbf{e}$ and 0 otherwise, so the chain rule makes $f(\mathbf{x}) = Pr'(\mathbf{x}, \mathbf{e})$ for the pruned network $\mathcal{N}'$, and Theorem 6.5 with query variables $\mathbf{Q}$ gives $Pr'(\mathbf{x}, \mathbf{e}) = Pr(\mathbf{x}, \mathbf{e})$. Both (I1) and (I2) reduce to this, $\mathbf{Y}$ being empty.
One iteration of Lines 6–8. Let $Z = \pi(i)$, $\mathbf{X}' = \mathbf{X} \setminus \{Z\}$ and $\mathbf{Y}' = \mathbf{Y} \cup \{Z\}$; let f' be the Line 6 product of the factors mentioning Z and g the product of the rest, so $f = f'g$ with g free of Z, and $f_{\text{new}} = (\max_Z f')g = \max_Z f$ by Theorem 10.1. Hence by (I2),

$$\begin{aligned} f_{\text{new}}(\mathbf{x}') &= \max_z \max_{\mathbf{y}} Pr(z, \mathbf{x}', \mathbf{y}, \mathbf{e}) \\ &= \max_{\mathbf{y}'} Pr(\mathbf{x}', \mathbf{y}', \mathbf{e}), \end{aligned}$$

the instantiations of $\mathbf{Y}'$ being exactly the pairs $(z, \mathbf{y})$; this is (I2) for f_{new}. Definition 10.2 at Line 7 and the product rule at Line 8 give

$$f_{\text{new}}[\mathbf{x}'] = z^\star\, f'[z^\star, \mathbf{x}']\, g[\mathbf{x}'] \;=\; z^\star\, f[z^\star, \mathbf{x}'],$$

where $z^\star = \operatorname{argmax}_z f'(z, \mathbf{x}')$; by induction $f[z^\star, \mathbf{x}']$ instantiates $\mathbf{Y}$, and $Z \notin \mathbf{Y}$, so this instantiates exactly $\mathbf{Y}'$. For the numeric half of (I1), two cases.
(i) $g(\mathbf{x}') > 0$: multiplying by the z-constant $g(\mathbf{x}')$ does not move the argmax, so $z^\star = \operatorname{argmax}_z f(z, \mathbf{x}')$, and (I1) for f at the row $(z^\star, \mathbf{x}')$ gives

$$\begin{aligned} Pr(\mathbf{x}', f_{\text{new}}[\mathbf{x}'], \mathbf{e}) &= Pr\big(\mathbf{x}', z^\star, f[z^\star, \mathbf{x}'], \mathbf{e}\big) \;=\; f(z^\star, \mathbf{x}') \\ &= \max_z f(z, \mathbf{x}') \;=\; f_{\text{new}}(\mathbf{x}'). \end{aligned}$$

(ii) $g(\mathbf{x}') = 0$: then $f(z, \mathbf{x}') = 0$ for every z, so $f_{\text{new}}(\mathbf{x}') = 0$, and by the (I2) part just proved every extension of $\mathbf{x}'\mathbf{e}$ has probability 0, the recorded witness included.
On return. After $|\mathbf{Q}|$ iterations $\mathbf{X} = \emptyset$ and $\mathbf{Y} = \mathbf{Q}$, so the returned factor is trivial and the invariant at $\top$ reads $f(\top) = \max_{\mathbf{q}} Pr(\mathbf{q}, \mathbf{e}) = Pr(f[\top], \mathbf{e})$, with $f[\top]$ an instantiation of all of $\mathbf{Q}$: an MPE instantiation for $\mathbf{e}$, of the returned probability. ■

Problem 10.11. Extend Algorithm VE MPE so it returns a structure Δ from which all MPE solutions can be enumerated in time linear in their count and linear in the size of Δ. The extended algorithm should have the same complexity as VE MPE. Hint: Appeal to NNF circuits.

Solution. Replace the single witness $f[\mathbf{x}]$ of Definition 10.2 by a pointer $\alpha_f(\mathbf{x})$ into a shared NNF circuit Δ over the literals $X = x$, built by three rules: at Line 4 set $\alpha_{h^{\mathbf{e}}}(\mathbf{x}) = \top$ for every reduced CPT; at Line 6 set $\alpha_{f_1 f_2}(\mathbf{z}) = \alpha_{f_1}(\mathbf{x}) \wedge \alpha_{f_2}(\mathbf{y})$ for the rows $\mathbf{x}, \mathbf{y} \sim \mathbf{z}$; and at Line 7, with $m = \max_z f(z, \mathbf{x}')$ and $Z^\star = \{z : f(z, \mathbf{x}') = m\}$ the set of *all* maximizers,

$$\alpha_{f_i}(\mathbf{x}') \;=\; \bigvee_{z \in Z^\star} \Big((Z = z) \wedge \alpha_f(z, \mathbf{x}') \Big).$$

Return $\Delta = \alpha_f(\top)$ for the trivial factor of Line 10.
Δ *is a smooth d-DNNF.* As in Exercise 10.10 each variable is recorded into exactly one factor when eliminated, so the variables mentioned by $\alpha_f(\mathbf{x})$ are exactly those eliminated into f, the same set for every row of f. Decomposability: α_{f_1} and α_{f_2} mention the disjoint sets eliminated into f_1 and f_2, while $(Z = z)$ mentions Z and $\alpha_f(z, \mathbf{x}')$ only variables eliminated before Z. Determinism: distinct disjuncts fix Z to distinct values. Smoothness: every disjunct mentions $\{Z\}$ together with the variables eliminated into f.
Invariant. For every row with $f(\mathbf{x}) > 0$,

$$\text{models}\big(\alpha_f(\mathbf{x})\big) \;=\; \{\mathbf{y} : Pr(\mathbf{x}, \mathbf{y}, \mathbf{e}) = f(\mathbf{x})\},$$

where $f(\mathbf{x}) = \max_{\mathbf{y}} Pr(\mathbf{x}, \mathbf{y}, \mathbf{e})$ by Exercise 10.10. Initially $\mathbf{Y} = \emptyset$ and $\alpha_f(\mathbf{x}) = \top$, so both sides are the singleton empty instantiation. For the step, let Z be the pivot, $f = f'g$ with g free of Z and $f_{\text{new}} = (\max_Z f')g$, and suppose $f_{\text{new}}(\mathbf{x}') > 0$; then $g(\mathbf{x}') > 0$, so scaling by it preserves maximizers, $Z^\star = \operatorname{argmax}_z f(z, \mathbf{x}')$, and the two rules give

$$\alpha_{f_{\text{new}}}(\mathbf{x}') \;=\; \bigvee_{z \in Z^\star} \Big((Z = z) \wedge \alpha_f(z, \mathbf{x}') \Big).$$

For $z \in Z^\star$ we have $f(z, \mathbf{x}') = f_{\text{new}}(\mathbf{x}') > 0$ and the hypothesis applies; for $z \notin Z^\star$, $\max_{\mathbf{y}} Pr(z, \mathbf{x}', \mathbf{y}, \mathbf{e}) = f(z, \mathbf{x}') < f_{\text{new}}(\mathbf{x}')$, so no extension through such a z attains the maximum and the omitted disjuncts lose nothing. The union over $Z^\star$ is thus exactly the set of instantiations $\mathbf{y}'$ of $\mathbf{Y} \cup \{Z\}$ attaining $f_{\text{new}}(\mathbf{x}')$. At Line 10, where $\mathbf{X} = \emptyset$ and $\mathbf{Y}$ is every network variable, the models of Δ are therefore precisely the MPE instantiations, provided $Pr(\mathbf{e}) > 0$.
Cost and enumeration. A multiplication adds one $\wedge$-node per row of the product and a maximization at most $|Z|$ $\wedge$-nodes and one $\vee$-node per row of the result, matching the row-touches the numeric algorithm already makes; so $|\Delta| = O(n \exp(w))$ and the running time stays $O(n \exp(w))$, the complexity of VE MPE. To enumerate, recurse: a literal and $\top$ yield themselves, an $\wedge$-node concatenates the models of its children, always consistently by decomposability, and a $\vee$-node unions them, disjointly by determinism and with every variable of the node already assigned by smoothness. The traversal never backtracks and never repeats a model, so it costs $O(|\Delta| \cdot \#\text{MPE})$. ■

Problem 10.12. Prove Lemma 10.1 on Page 267, which states: Let $\mathcal{N}'$ be the network resulting from splitting variables in network $\mathcal{N}$. Let Pr' and Pr be their corresponding distributions. We then have

$$Pr(\mathbf{x}) \;=\; \beta \cdot Pr'(\mathbf{x}, \hat{\mathbf{x}}),$$

where $\mathbf{X}$ are the variables of network $\mathcal{N}$, $\hat{\mathbf{x}}$ is an instantiation of all clone variables that is implied by $\mathbf{x}$, and β is the split cardinality.
Recall Definition 10.3: splitting node X according to children $\mathbf{Z} \subseteq \text{children}(X)$ removes the edges $X \to Z$ for $Z \in \mathbf{Z}$, adds a new root node $\hat{X}$ with the same values as X and with the nodes $\mathbf{Z}$ as children, assigns $\hat{X}$ a uniform CPT, and replaces X by $\hat{X}$ in the CPT of each node in $\mathbf{Z}$. The split cardinality β is the number of instantiations of the split (clone) variables. The instantiation $\hat{\mathbf{x}}$ implied by $\mathbf{x}$ assigns each clone $\hat{X}$ the value that $\mathbf{x}$ assigns to X.

Solution. In the chain rule for $\mathcal{N}'$ every factor but the clone prior is literally a factor of $\mathcal{N}$, and the clone prior is $1/\beta$.
One split. Let $\mathcal{N}'$ split X according to $\mathbf{Z}$, with clone $\hat{X}$, so $\beta = |\text{val}(X)|$. Fix $\mathbf{x}$, let x be the value it assigns X, so that $\hat{\mathbf{x}} : \hat{X} = x$. The chain rule in $\mathcal{N}'$ gives

$$Pr'(\mathbf{x}, \hat{\mathbf{x}}) \;=\; \theta'_{\hat{x}} \cdot \prod_{V \in \mathbf{X}} \theta'_{v | \mathbf{u}'_V},$$

with $v, \mathbf{u}'_V$ the values $\mathbf{x}\hat{\mathbf{x}}$ assigns V and its parents in $\mathcal{N}'$. Here $\theta'_{\hat{x}} = 1/\beta$, the clone being a root with a uniform CPT over $|\text{val}(X)|$ values; and every other factor is unchanged. Indeed for $V \notin \mathbf{Z}$ Definition 10.3 touches neither the parents nor the CPT of V, the case $V = X$ included, splitting removing only outgoing edges of X; and for $V \in \mathbf{Z}$ the parents become $(\mathbf{U}_V \setminus \{X\}) \cup \{\hat{X}\}$ with the old CPT column X renamed $\hat{X}$, while $\mathbf{x}\hat{\mathbf{x}}$ assigns $\hat{X}$ the same value x, so the selected row is the same one as before. Hence $Pr'(\mathbf{x}, \hat{\mathbf{x}}) = \frac{1}{\beta} Pr(\mathbf{x})$ by the chain rule in $\mathcal{N}$.
Several splits. Induct on the number k of splits; $k = 0$ is trivial, with $\beta = 1$, and $k = 1$ is the above. Let $\mathcal{N}''$ be the network after the first $k - 1$ splits, with clones $\hat{\mathbf{C}}$ and split cardinality β'', so that $Pr(\mathbf{x}) = \beta'' Pr''(\mathbf{x}, \hat{\mathbf{c}})$ for the implied $\hat{\mathbf{c}}$. The last split is of some node Y of $\mathcal{N}''$, so $\beta = \beta'' |\text{val}(Y)|$, and the single-split case applied at the instantiation $\mathbf{x}\hat{\mathbf{c}}$ gives

$$Pr''(\mathbf{x}, \hat{\mathbf{c}}) \;=\; |\text{val}(Y)| \cdot Pr'\big(\mathbf{x}, \hat{\mathbf{c}}, \hat{Y} = y\big),$$

y being the value $\mathbf{x}\hat{\mathbf{c}}$ assigns Y. Finally $(\hat{\mathbf{c}}, \hat{Y} = y)$ is exactly the clone instantiation $\hat{\mathbf{x}}$ of $\mathcal{N}'$ implied by

$\mathbf{x}$: immediate if Y is an original variable, and if Y is itself a clone of an original C then $\hat{\mathbf{c}}$ already sets Y to the value $\mathbf{x}$ gives C, which $\hat{Y}$ inherits. Multiplying the two displays, $Pr(\mathbf{x}) = \beta \cdot Pr'(\mathbf{x}, \hat{\mathbf{x}})$. ■

Problem 10.13. Let $\mathcal{N}$ be a network with treewidth w and let $\mathcal{N}'$ be another network that results from splitting m variables in $\mathcal{N}$. Show that $w' \geq w - m$, where w' is the treewidth of network $\mathcal{N}'$. This means that to reduce the treewidth of network $\mathcal{N}$ by m, we must split at least m variables.

Solution. Equivalently $w \leq w' + m$, which follows by turning an optimal jointree for $\mathcal{N}'$ into one for $\mathcal{N}$ at a cost of m nodes per cluster; treewidth is the width of the best jointree by Definitions 9.2 and 9.13. Let $X_1, \ldots, X_m$ be the split variables with clones $\hat{X}_1, \ldots, \hat{X}_m$, let $(T, \mathbf{C}')$ be a jointree for $\mathcal{N}'$ of width w', and on the same tree put

$$\mathbf{C}_i \;=\; \big(\mathbf{C}'_i \setminus \{\hat{X}_1, \ldots, \hat{X}_m\}\big) \;\cup\; \{X_1, \ldots, X_m\} :$$

delete every clone from every cluster, add every split variable to every cluster. For cascaded splits, where a split variable may itself be a clone, read $\{X_1, \ldots, X_m\}$ as $\varphi(\{X_1, \ldots, X_m\})$ with φ sending a clone to the node of $\mathcal{N}$ it ultimately clones, a set of at most m nodes; everything below survives that reading, since a clone parent $\hat{X}$ of V in $\mathcal{N}'$ always has $\varphi(\hat{X}) \in \mathbf{U}_V$. Either way $|\mathbf{C}_i| \leq |\mathbf{C}'_i| + m \leq w' + 1 + m$. The three conditions of Definition 9.13 hold for $(T, \mathbf{C})$. Condition 1 is immediate, the clones having been deleted and the X_j being nodes of $\mathcal{N}$. For condition 2, splitting replaces a parent only by that parent's clone and adds no non-clone parent, so

$$\text{family}_{\mathcal{N}'}(V) \;\subseteq\; \big(\text{family}_{\mathcal{N}}(V) \setminus \{X_1, \ldots, X_m\}\big) \cup \{\hat{X}_1, \ldots, \hat{X}_m\},$$

and a cluster $\mathbf{C}'_i$ covering the left-hand side therefore contains $\text{family}_{\mathcal{N}}(V) \setminus \{X_1, \ldots, X_m\}$, to which $\mathbf{C}_i$ adds back every X_j; hence $\mathbf{C}_i \supseteq \text{family}_{\mathcal{N}}(V)$, the case $V = X_j$ included, splitting removing only outgoing edges of X_j and so leaving its family alone. For condition 3: (i) each X_j lies in every cluster, and the whole of T is connected; (ii) any other V is neither a split variable nor a clone, so $V \in \mathbf{C}_i$ exactly when $V \in \mathbf{C}'_i$, and connectedness is the jointree property of $(T, \mathbf{C}')$.
So $\mathcal{N}$ has a jointree of width at most $w' + m$, whence $w \leq w' + m$ and $w' \geq w - m$. ■

Problem 10.14. Let $\mathcal{N}$ be a Bayesian network and let $\mathbf{C}$ be a corresponding loop cutset. Provide tight lower and upper bounds on the treewidth of a network that results from splitting variables $\mathbf{C}$ in $\mathcal{N}$. Note: A variable $C \in \mathbf{C}$ can be split according to any number of children.
Recall Definition 8.1: a set of nodes $\mathbf{C}$ is a loop cutset for $\mathcal{N}$ if removing the edges outgoing from nodes $\mathbf{C}$ renders the network a polytree.

Solution. $k \leq w' \leq w + c$, where k is the largest number of parents of a node of $\mathcal{N}$, w its treewidth and c the number of clones introduced; and $w' \leq w$ when every split is an edge deletion or an all-children split, in any sequence. Both pairs are tight. Note $k \leq w$ always, the family of a node with k parents being a clique of size $k + 1$ in the moral graph (Definitions 9.1 and 9.2), so Theorem 9.1 applies.
Lower bound. Splitting changes nobody's parent count: Definition 10.3 replaces each edge $C \to Z$, $Z \in \mathbf{Z}$, by $\hat{C} \to Z$, leaves the parents of C alone and makes $\hat{C}$ a root. So the moral graph of $\mathcal{N}'$ still holds a clique of size $k + 1$ and $w' \geq k$, for any split of any network.
It is attained. Split each $C \in \mathbf{C}$ according to each of its children separately, one clone $\hat{C}_Y$ per child Y. The result is $\mathcal{N}_{\mathbf{C}}$, which is a polytree because $\mathbf{C}$ is a loop cutset (Definition 8.1), with the clones added one at a time as fresh degree-one nodes, which cannot create an undirected cycle; so $\mathcal{N}'$ is a polytree. Its family clusters form a jointree — one tree node per network node V with cluster $\{V\} \cup \mathbf{U}_V$, joined to the clusters of the children of V, disconnected components joined arbitrarily since they share no variable — because the clusters containing V are that of V together with those of its children, all adjacent to it. Its width is the largest family size minus one, still k, so $w' = k$.
$w' \leq w$ for the two standard shapes. One split of either shape maps a jointree into a jointree of no larger width, and since both constructions accept an arbitrary jointree of an arbitrary network they

compose. Let $(T, \mathbf{K})$ be a jointree of $\mathcal{N}$ of width w.
(a) *C split according to a single child Y.* Pick i with $\text{family}_{\mathcal{N}}(Y) \subseteq \mathbf{K}_i$ and hang a new node i^* off i with

$$\mathbf{K}_{i^*} \;=\; \big(\text{family}_{\mathcal{N}}(Y) \setminus \{C\}\big) \cup \{\hat{C}\},$$

of size $|\text{family}_{\mathcal{N}}(Y)| \leq w+1$. It is exactly $\text{family}_{\mathcal{N}'}(Y)$ and contains $\text{family}_{\mathcal{N}'}(\hat{C}) = \{\hat{C}\}$, every other family being unchanged. The jointree property survives: $\hat{C}$ lies in i^* alone, every other member of $\mathbf{K}_{i^*}$ lies also in the adjacent $\mathbf{K}_i$, and $C \notin \mathbf{K}_{i^*}$.
(b) *C split according to all its children with one clone.* Rename C to $\hat{C}$ in every cluster, which covers every family of $\mathcal{N}'$ except $\{C\} \cup \mathbf{U}_C$, each child of C keeping exactly its renamed old family. Hang $\{C\} \cup \mathbf{U}_C$, of size $\leq w+1$, off a cluster containing $\{\hat{C}\} \cup \mathbf{U}_C$, which exists because that cluster covered $\text{family}_{\mathcal{N}}(C)$ before the renaming; the jointree property survives as in (a).
It is attained. When the cutset consists of roots and each C is split by shape (b), $\mathcal{N}'$ is $\mathcal{N}$ with each C renamed $\hat{C}$ plus $|\mathbf{C}|$ isolated nodes, which do not affect treewidth, so $w' = w$: for instance the diamond $A \to B$, $A \to C$, $B \to D$, $C \to D$, whose loop cutset is $\{A\}$ and whose treewidth is 2 before and after.
Why w fails in general. Definition 10.3 also allows a clone to inherit a proper subset of two or more children. For

$$\mathcal{N}: \quad A \to B, \quad A \to C, \quad B \to C, \quad A \to D, \quad C \to D, \quad A \to E, \quad D \to E,$$

deleting the edges out of A leaves the chain $B \to C \to D \to E$, so $\{A\}$ is a loop cutset; the families $\{B,A\},\{C,A,B\},\{D,A,C\},\{E,A,D\}$ moralize to three triangles glued at A, and the jointree $ABC - ACD - ADE$ gives $w = 2$. Splitting A according to $\{B, D\}$ leaves the families $\{B,\hat{A}\},\{C,A,B\},\{D,\hat{A},C\},\{E,A,D\}$, whose moral graph contains the four-cycle $B - \hat{A} - D - A - B$ (neither $B - D$ nor $A - \hat{A}$ is an edge) with C adjacent to all four of its nodes: a four-spoke wheel of minimum degree 3. A graph of minimum degree d has treewidth at least d, the first node eliminated contributing a cluster of itself and all its neighbours (Definition 9.4), and treewidth never increases when nodes or edges are deleted, so $w' \geq 3 > w$; the order $E, B, \hat{A}, A, C, D$ has width 3, so $w' = 3$ exactly.
$w' \leq w + c$. A single split raises the treewidth by at most one: add $\hat{X}$ to every cluster containing X. Those clusters form a connected subtree, so the jointree property holds for $\hat{X}$; each new family $(\text{family}_{\mathcal{N}}(Y) \setminus \{X\}) \cup \{\hat{X}\}$, $Y \in \mathbf{Z}$, is covered at the cluster that covered $\text{family}_{\mathcal{N}}(Y)$, which contains X; and no cluster grew by more than one node. Iterating over the c clones gives the bound, for every split of every network.
It is attained for every c. First a tool: if disjoint connected sets $\mathbf{S}_1, \ldots, \mathbf{S}_r$ of moral-graph nodes are pairwise linked by a moral edge, the treewidth is at least $r-1$. Take an optimal jointree, whose clusters cover every family and hence every moral edge, and replace in every cluster each node of $\mathbf{S}_i$ by one token s_i; clusters do not grow. The clusters holding s_i are the union of the subtrees of the nodes of $\mathbf{S}_i$, itself a subtree, since for each edge inside $\mathbf{S}_i$ the two endpoint subtrees share the cluster covering that edge and $\mathbf{S}_i$ is connected; and each linking edge puts s_i and s_j in one cluster. The result is a jointree of no larger width for a graph in which $s_1, \ldots, s_r$ is a clique, so Theorem 9.1 gives width $\geq r-1$.
Now fix $t \geq 1$, let $n = (t+1)(t+3)$, and let $\mathcal{F}_t$ have nodes $A, B_1, \ldots, B_n$ with edges $A \to B_j$ for all j and $B_j \to B_{j+1}$ for $j < n$. Deleting the edges out of A leaves the chain, so $\{A\}$ is a loop cutset; moralization adds no edge, the parents $\{A, B_j\}$ of B_{j+1} being already adjacent, and $w = 2$ — the triangle A, B_1, B_2 below, the order $B_1, \ldots, B_n, A$ with clusters $\{B_j, B_{j+1}, A\}$ above. Split A according to $\mathbf{Z}_i = \{B_j : j \equiv i \pmod{t+1}\}$ with clone $\hat{A}_i$, for $i = 1, \ldots, t$ in turn; this is legal, since before the i-th split the children of A are the classes $i, \ldots, t, 0$, of which $\mathbf{Z}_i$ is a proper subset. Each B_j then has two parents, its owner ($\hat{A}_i$ when $j \equiv i$ for $1 \leq i \leq t$, else A) and its predecessor B_{j-1}, so the moral graph is the chain $B_1 - \cdots - B_n$ together with, for each j, the edges from the owner of B_j to B_j and to B_{j-1}; no two owners are adjacent, sharing no child. Cut the chain into $t+3$ consecutive blocks $\mathbf{P}_1, \ldots, \mathbf{P}_{t+3}$ of $t+1$ nodes, each meeting every residue class, so every owner has a neighbour in every block. The sets

$$\mathbf{S}_i = \{\hat{A}_i\} \cup \mathbf{P}_i \;(i = 1, \ldots, t), \qquad \mathbf{S}_{t+1} = \{A\} \cup \mathbf{P}_{t+1}, \qquad \mathbf{S}_{t+2} = \mathbf{P}_{t+2}, \qquad \mathbf{S}_{t+3} = \mathbf{P}_{t+3}$$

are disjoint and connected, blocks being subpaths and each owner adjoining its own block, and pairwise linked: a set holding an owner links to every other through that owner's neighbour there, and $\mathbf{S}_{t+2}$ links to $\mathbf{S}_{t+3}$ by the chain edge between consecutive blocks. Hence $w' \geq (t+3)-1 = t+2$, while $c = t$ clones and the unconditional bound give $w' \leq w + c = t+2$: equality for every $c \geq 1$. ■

9.3 Exercises 10.15–10.21

Problem 10.15. Suppose that we have a Bayesian network N with a corresponding jointree J. Let N' be a network that results from splitting some variable X in N according to its children (i.e., one clone is introduced). Show how we can obtain a jointree for network N' by modifying the clusters of jointree J. Aim for the best jointree possible (minimize its width). Show how this technique can be used to develop a greedy method for obtaining a split network that has a particular width.

Solution. Keep X only where its own CPT is hosted, rename it to $\hat{X}$ on a smallest subtree carrying the hosts of the children's CPTs, and hang XU off as a fresh leaf; the width never increases. Let U be the parents of X and $Y_1, \ldots, Y_k$ its children, so by Definition 10.3 the families of N' are XU, the singleton $\{\hat{X}\}$, the $(\text{fam}_N(Y_i) \setminus \{X\}) \cup \{\hat{X}\}$, and $\text{fam}_N(V)$ for every other V.
Construction. Let T_X be the clusters of J containing X; it is a connected subtree hosting each of XU and the $\text{fam}_N(Y_i)$.

1. For each Y_i pick a host $h(Y_i) \in T_X$ with $\text{fam}_N(Y_i) \subseteq h(Y_i)$, and put $H = \{h(Y_1), \ldots, h(Y_k)\}$.
2. Let $\hat{T}$ be the smallest connected subtree containing H, the union of the paths between members of H; it lies inside T_X, which is itself connected and contains H.
3. Replace X by $\hat{X}$ in every cluster of $\hat{T}$, and delete X from every cluster of $T_X \setminus \hat{T}$.
4. Attach a fresh leaf $C_{\text{new}} = XU$ to any cluster $C_a \supseteq XU$ of J, which exists since J hosts the CPT of X; the separator is U.

Validity. X now occupies C_{new} alone and $\hat{X}$ the connected $\hat{T}$; any other variable V keeps exactly its old clusters, plus the leaf C_{new} when $V \in U \subseteq C_a$, and a leaf hung on a member of a connected subtree leaves it connected. As for families, $XU = C_{\text{new}}$; $\{\hat{X}\}$ lies in any cluster of $\hat{T}$, which is nonempty as X has a child; $\text{fam}_{N'}(Y_i)$ lies in the relabelled $h(Y_i) \in \hat{T}$; and every remaining family lost only X, which belongs to no other family of N'.
Width. Step 3 only renames or deletes, so no old cluster grows, and $|C_{\text{new}}| = |U| + 1 \leq w + 1$ because $C_a \supseteq XU$ already; hence $\text{width}(J') \leq \text{width}(J)$, strictly so exactly when every maximum-size cluster of J lies in $T_X \setminus \hat{T}$ and $|U| + 1 \leq w$. Taking the Steiner subtree $\hat{T}$ rather than all of T_X is what strips $\hat{X}$ out of as many clusters as the jointree property permits; the one remaining freedom is the choice of hosts in step 1, which can be searched greedily so that $\hat{T}$ avoids the large clusters.
Greedy method. Fix an affordable width $w^\star$, build any jointree J for N and set $N' \leftarrow N$. While $\text{width}(J) > w^\star$: for each variable X of N' occurring in a maximum-size cluster of J but not heading the family that fills it, compute J_X by the construction above; score X by $\Delta(X)/\log|X|$ with $\Delta(X) = \text{width}(J) - \text{width}(J_X)$, breaking $\Delta = 0$ ties by the drop in the number of maximum-size clusters; then commit the best candidate, splitting it in N' and setting $J \leftarrow J_X$. Return N' with J. Each iteration is local — find T_X, take the Steiner subtree, relabel, prune, add one leaf — so no jointree is ever rebuilt and the new width is read off directly, while the $\log|X|$ divisor charges each candidate the factor $|X|$ it contributes to the split cardinality β of Theorem 10.2. The loop terminates, every committed split lowering either the width or the number of maximum-size clusters, both nonnegative integers; it may also halt short of $w^\star$, since splitting changes no family size and so cannot push the width below $\max_V |\text{fam}_N(V)| - 1$.

Problem 10.16. Consider a Bayesian network N, a corresponding jointree J, and assume that the CPTs for network N have been assigned to clusters in J. Consider a separator S_{ij} that contains variable X and suppose that the CPT of variable X has been assigned to a cluster on the j-side of edge $i-j$. Consider now a transformation that produces a new jointree J' as follows:

- Replace all occurrences of variable X by $\hat{X}$ on the i-side of edge $i-j$ in the jointree.

- Remove all occurrences of X and $\hat{X}$ from clusters that have not been assigned CPTs mentioning X, as long as the removal will not destroy the jointree property.

Describe a network N' that results from splitting variables in N for which J' would be a valid jointree.

Solution. J' is a valid jointree for the network N' obtained from N by splitting X according to the children

$$Z = \{\, Y \in \mathbf{Y} : \text{the CPT of } Y \text{ is assigned to an } i\text{-side cluster} \,\},$$

in the sense of Definition 10.3; the split cardinality is $\beta = |X|$, so Theorem 10.2 reads $\mathrm{MPE}_P(e) \leq |X| \cdot \mathrm{MPE}'_P(e, \hat{e})$. Write U for the parents of X and $\mathbf{Y}$ for its children, and call the two components of the jointree minus the edge $i-j$ the i-side and the j-side; each CPT sits in exactly one cluster, so Z is well defined.

Why. Let T_X be the clusters containing X, connected by the jointree property, and let T_X^i, T_X^j be its intersections with the two sides; intersections of connected subtrees are connected, and both are nonempty because $X \in S_{ij} = C_i \cap C_j$ puts C_i and C_j in T_X. The first bullet renames X to $\hat{X}$ in exactly the clusters of T_X^i, so afterwards X occupies T_X^j and $\hat{X}$ occupies T_X^i, both connected, and no other variable is touched. The families of N' are covered: XU by the j-side cluster hosting the CPT of X, which the rename leaves alone; for $Y \in Z$, by the i-side host $C \supseteq \mathrm{fam}_N(Y)$, which has $X \in C$, hence $C \in T_X^i$, so the rename turns it into a cluster containing $(\mathrm{fam}_N(Y) \setminus \{X\}) \cup \{\hat{X}\}$; for $Y \in \mathbf{Y} \setminus Z$, by its untouched j-side host; $\{\hat{X}\}$ by C_i, renamed since $X \in S_{ij} \subseteq C_i$; and every remaining family by its old host, those families not mentioning X at all. The second bullet deletes X or $\hat{X}$ only from clusters hosting no CPT that mentions X, and only when connectivity survives, so it breaks neither property; as deletions only shrink clusters, $\mathrm{width}(J') \leq \mathrm{width}(J)$.

Degenerate case. If $Z = \emptyset$, then N' is N with a barren root $\hat{X}$ added. The second bullet cannot strip $\hat{X}$ from every i-side cluster, since the family $\{\hat{X}\}$ would lose its host and the jointree property would fail; one cluster keeps the clone.

Problem 10.17. Let N' be a network that results from splitting nodes in network N, and let $\Pr'$ and $\Pr$ be their corresponding distributions. Show that

$$\Pr(e) \leq \beta \cdot \Pr'(e, \hat{e}),$$

where $\hat{e}$ is the instantiation of clone variables that is implied by evidence e and β is the split cardinality.

Solution. Lemma 10.1, proved in Exercise 10.12, gives $\Pr'(x, \hat{x}) = \Pr(x)/\beta$ for every instantiation x of the variables X of N, where $\hat{x}$ is the clone instantiation agreeing with x and $\beta = \prod_{i=1}^{m} |X_i|$ over the split variables. Hence

$$\Pr(e) \;=\; \sum_{x \sim e} \Pr(x) \;=\; \beta \sum_{x \sim e} \Pr'(x, \hat{x}).$$

Every term of that sum is a term of

$$\Pr'(e, \hat{e}) \;=\; \sum_{(x, \hat{s}) \sim e\hat{e}} \Pr'(x, \hat{s}).$$

Indeed $(x, \hat{x}) \sim e\hat{e}$ whenever $x \sim e$: for a split variable X_i instantiated by e, both $\hat{e}$ and $\hat{x}$ assign $\hat{X}_i$ the value e gives X_i, since $x \sim e$, while clones left free by $\hat{e}$ constrain nothing. The map $x \mapsto (x, \hat{x})$ is injective, so distinct terms stay distinct, and all terms are nonnegative; therefore $\sum_{x \sim e} \Pr'(x, \hat{x}) \leq \Pr'(e, \hat{e})$ and

$$\Pr(e) \;\leq\; \beta \cdot \Pr'(e, \hat{e}). \qquad \blacksquare$$

Problem 10.18. Prove Theorem 10.3.
For reference, Theorem 10.2 reads: let N' be the network that results from splitting variables in network N and let β be the split cardinality; for evidence e on network N, let $\hat{e}$ be the instantiation of split variables that is implied by e; then $\mathrm{MPE}_P(e) \leq \beta \cdot \mathrm{MPE}'_P(e, \hat{e})$, where $\mathrm{MPE}_P(e)$ and $\mathrm{MPE}'_P(e, \hat{e})$ are MPE probabilities with respect to the networks N and N'.
Theorem 10.3 reads: consider Theorem 10.2 and suppose that distributions Pr and Pr$'$ are induced by networks N and N', respectively. If the evidence e includes all split variables and if Q are the variables of N not in E, then

$$\Pr(Q, e) \;=\; \beta \cdot \Pr'(Q, e, \hat{e}).$$

Solution. When e instantiates every split variable, the clone instantiation is forced to agree with the original and Lemma 10.1 applies term by term. Fix an instantiation q of $Q = X \setminus E$, where X are the variables of N. Since Q and E partition X, the instantiation $x = qe$ is full and $\Pr(q, e) = \Pr(x)$ is a single term of the joint rather than a sum. Since $S \subseteq E$, the instantiation $\hat{e}$ fixes every clone, to the value e gives the corresponding split variable; as x extends e, that is the value x gives it, so $\hat{e} = \hat{x}$ in the notation of Lemma 10.1. Hence $(q, e, \hat{e}) = (x, \hat{x})$ is a full instantiation of the variables $X \cup \hat{S}$ of N', and Lemma 10.1 — proved in Exercise 10.12, with $\beta = \prod_{i=1}^{m} |X_i|$ — gives

$$\Pr'(q, e, \hat{e}) \;=\; \Pr'(x, \hat{x}) \;=\; \frac{1}{\beta} \Pr(x) \;=\; \frac{1}{\beta} \Pr(q, e).$$

As q was arbitrary, $\Pr(Q, e) = \beta \cdot \Pr'(Q, e, \hat{e})$ as factors over Q. ■

Problem 10.19. Consider the classical jointree algorithm as defined by Equations 7.4 and 7.5 in Chapter 7:

$$M_{ij} \;=\; \sum_{C_i \setminus S_{ij}} \Phi_i \prod_{k \neq j} M_{ki}, \qquad f_i(C_i) \;=\; \Phi_i \prod_{k} M_{ki}.$$

Suppose that we replace all summations by maximization when computing messages,

$$M_{ij} \;=\; \max_{C_i \setminus S_{ij}} \Phi_i \prod_{k \neq j} M_{ki}.$$

What are the semantics of the factor $f_i(C_i)$ in this case? In particular, can we use it to recover answers to MPE queries?

Solution. The factor $f_i(C_i)$ becomes the *max-marginal* $\mathrm{MPE}_P(C_i e)$ in place of the marginal $\Pr(C_i, e)$, and yes: the MPE probability, the per-variable max-marginals and an MPE instantiation all follow from one propagation.
Notation. Let Φ_i be the product of the CPTs and evidence indicators assigned to cluster i, so $\prod_i \Phi_i = \Pr(X)\lambda_e(X)$ over the network variables $X = \bigcup_i C_i$; for an edge $i - j$ let $T_{i \to j}$ be the component of i when it is deleted, and put $V_{i \to j} = \bigcup_{k \in T_{i \to j}} C_k$ and $\Phi_{i \to j} = \prod_{k \in T_{i \to j}} \Phi_k$. The jointree property gives the separation fact $V_{k \to i} \cap C_i = S_{ki}$: a variable of $V_{k \to i}$ occurring also outside $T_{k \to i}$ has its connected cluster subtree crossing the edge $k - i$, so it lies in $C_k \cap C_i$. Hence the variables of $V_{k \to i} \setminus S_{ki}$ occur in no factor outside that branch, and distinct branches at i are variable-disjoint outside C_i.
Messages. $M_{ij} = \max_{V_{i \to j} \setminus S_{ij}} \Phi_{i \to j}$, by induction on $|T_{i \to j}|$. For $T_{i \to j} = \{i\}$ this is the definition. Otherwise $\Phi_{i \to j} = \Phi_i \prod_{k \neq j} \Phi_{k \to i}$ and, by the separation fact,

$$V_{i \to j} \setminus S_{ij} \;=\; (C_i \setminus S_{ij}) \;\uplus\; \biguplus_{k \neq j} (V_{k \to i} \setminus S_{ki}),$$

so, maximization being commutative (Definition 10.1),

$$\max_{V_{i\to j}\setminus S_{ij}} \Phi_{i\to j} = \max_{C_i\setminus S_{ij}} \Phi_i \prod_{k\neq j} \Big(\max_{V_{k\to i}\setminus S_{ki}} \Phi_{k\to i} \Big)$$
$$= \max_{C_i\setminus S_{ij}} \Phi_i \prod_{k\neq j} M_{ki} \;=\; M_{ij},$$

the first equality by Theorem 10.1 applied repeatedly, each variable of $V_{k\to i} \setminus S_{ki}$ occurring in $\Phi_{k\to i}$ and in no other factor of the product, and the second by the inductive hypothesis.
Semantics. The same computation at cluster i with every neighbour included gives

$$f_i(C_i) \;=\; \Phi_i \prod_k M_{ki} \;=\; \max_{X\setminus C_i} \Pr(X)\lambda_e(X),$$

that is, $f_i(c_i) = \max_{x\sim c_i e} \Pr(x) = \mathrm{MPE}_P(c_i e)$ when $c_i \sim e$ and 0 otherwise. This is not a marginal: in general $\sum_{c_i} f_i(c_i) \neq \Pr(e)$, and normalizing f_i means nothing.
Recovery. (i) $\max_{c_i} f_i(c_i) = \max_{x\sim e} \Pr(x) = \mathrm{MPE}_P(e)$, the same value at every cluster. (ii) Maximizing f_i over $C_i \setminus \{V\}$, at any cluster containing V, gives the factor $\mathrm{MPE}_P(V, e)$, so one propagation answers this for every variable at once. (iii) If the MPE instantiation $x^\star$ is unique then $f_i(c_i) = \mathrm{MPE}_P(e)$ exactly when $c_i = x^\star \downarrow C_i$, so the local arg max at each cluster is unique, the choices agree on separators, and $x^\star = \bigcup_i c_i^\star$. Under ties independent local choices may lock onto projections of different MPE instantiations and need not even cohere, so trace back instead: root at any r, pick $c_r^\star \in \arg\max_{c_r} f_r(c_r)$, and at a cluster j whose parent i is already fixed pick $c_j^\star$ maximizing f_j among the c_j agreeing with $c_i^\star$ on S_{ij}. These are consistent by construction and attain $\mathrm{MPE}_P(e)$: writing $s_{ij} = c_i^\star \downarrow S_{ij}$, the message formula gives $f_j(c_j) = M_{ij}(s_{ij}) \cdot \Phi_j(c_j) \prod_{k\neq i} M_{kj}(c_j)$ for every candidate c_j, so the constant $M_{ij}(s_{ij})$ drops out and the local choice realises $M_{ji}(s_{ij})$, the branch maximum the running total already charges; and by the separation fact the clusters fixed so far share with the j-branch only the variables of S_{ij}, so no earlier commitment is disturbed. Alternatively run the propagation on extended factors (Definition 10.2), so that each c_i carries an instantiation of $X \setminus C_i$ attaining $\mathrm{MPE}_P(c_i e)$ and the best entry of any single cluster yields an MPE instantiation outright.

Problem 10.20. Consider the classical jointree algorithm as defined by Equations 7.4 and 7.5 in Chapter 7:

$$M_{ij} \;=\; \sum_{C_i\setminus S_{ij}} \Phi_i \prod_{k\neq j} M_{ki}, \qquad f_i(C_i) \;=\; \Phi_i \prod_k M_{ki}.$$

Suppose that we replace all summations over MAP variables M by maximization when computing messages as follows:

$$M_{ij} \;=\; \max_{(C_i\setminus S_{ij})\cap M} \; \sum_{(C_i\setminus S_{ij})\setminus M} \Phi_i \prod_{k\neq j} M_{ki}.$$

What are the semantics of the factor $f_i(C_i)$ in this case? In particular, can we use it to recover answers relating to MAP queries?

Solution. $f_i(C_i)$ is not the MAP marginal but an upper bound on it: for every instantiation c_i of C_i,

$$\mathrm{MAP}_P(M \setminus C_i,\; c_i e) \;\leq\; f_i(c_i) \;\leq\; \Pr(c_i, e).$$

For $f_i = \Phi_i \prod_k M_{ki}$ is exactly the factor $\mathrm{VE}_{\mathrm{MAP}}$ (Algorithm 30) returns with MAP variables $M \setminus C_i$, evidence $c_i e$, and the elimination order π_i that the jointree rooted at cluster i induces (Section 9.3.5), each variable being eliminated in the cluster nearest the root that contains it; the message M_{ki} localises the eliminations of its own branch, legitimately, by the separation fact of Exercise 10.19. That order is in general *not* M-constrained (Definition 10.4): a MAP variable in a leaf cluster is maximized out before non-MAP variables nearer the root. So both inequalities are Theorem 10.5 with c_i folded into the evidence – the left one because π_i has moved maximizations inward past summations relative to a constrained order, which by Theorem 10.4 can only increase the value; the right one because

$\max_Y g \le \sum_Y g$ for nonnegative g, and the all-summation computation returns $\Pr(c_i, e)$ by Equation 7.5. It is exact exactly when π_i is M-constrained on $X \setminus C_i$, in particular when $M \subseteq C_i$ or $X \setminus C_i \subseteq M$. For a concrete gap, take the chain $A \to B \to C \to D$ over binary variables with the parameters

parent value	$\Pr(A{=}\text{true})$	$\Pr(B{=}\text{true} \mid A)$	$\Pr(C{=}\text{true} \mid B)$	$\Pr(D{=}\text{true} \mid C)$
true	.6	.2	.9	.3
false	–	.7	.4	.8

(the root A has no parent, so its single parameter is written in the first row) and the chain of clusters $C_1 = AB$, $C_2 = BC$, $C_3 = CD$, with $M = \{A, C\}$ and $e : D = \text{true}$. Enumeration gives $\Pr(e) = .5$ and $\text{MAP}_P(M, e) = .24$ at $A = \text{true}, C = \text{false}$, while rooting at C_1 (order C, B, A) returns $q_1 = .2628$ and rooting at C_2 or C_3 (order A, B, C) returns $q_2 = q_3 = .2528$ (Check!), so

$$\text{MAP}_P(M, e) = .24 \;\le\; .2528 \;\le\; .2628 \;\le\; \Pr(e) = .5;$$

no root is exact here, since every root maximizes a MAP variable before B is summed.

Three MAP answers do follow, all from one propagation.

(i) A global bound: $q_i = \max_{C_i \cap M} \sum_{C_i \setminus M} f_i(C_i)$ is what VE_{MAP} returns for MAP variables M and evidence e under π_i extended by $C_i \setminus M$ then $C_i \cap M$, so $\text{MAP}_P(M, e) \le q_i \le \Pr(e)$ by Theorem 10.5. This is the $\text{MAP}^u_P(X, i)$ that Line 3 of Algorithm 31 needs, one inward-outward propagation supplies it at every cluster at cost $O(n \exp(w))$, and $\min_i q_i$ is therefore available too.

(ii) Per-value bounds: with $X{=}x$ added to the evidence the same argument makes

$$B_x \;=\; \max_{(C_i \cap M) \setminus \{X\}} \sum_{C_i \setminus M} f_i(C_i)\Big|_{X=x}$$

an upper bound on $\text{MAP}_P(M \setminus \{X\}, e\, X{=}x)$, for every remaining MAP variable and every value at once, with $q_i = \max_x B_x$. This is the efficiency claim used in Exercise 10.21.

(iii) Exact MAP when $M \subseteq C_r$ for some cluster: then π_r is M-constrained, $\text{MAP}_P(M, e) = \max_{C_r \cap M} \sum_{C_r \setminus M} f_r(C_r)$, and the maximizing m is read off directly – possible only within the constrained treewidth of Definition 10.4.

The bound tightens under the promotion of Figure 10.11: moving a MAP variable from C_k into its root-ward neighbour C_i, subject to $|C_i| \le w + 1$, preserves the jointree properties and the width but postpones that variable's elimination past the non-MAP eliminations at C_i.

Problem 10.21. Consider Line 3 of Algorithm 31 and let

$$B_x \;=\; \text{MAP}^u_P(\mathbf{X} \setminus \{X\}, ix).$$

Here $\mathbf{X}$ is the set of variables still to be instantiated at the current search node and X is a single variable of that set. For reference, Algorithm 31, $\text{BB}_{\text{MAP}}(N, M, e)$, takes a Bayesian network N, MAP variables M, and evidence e with $E \cap M = \emptyset$, and returns a MAP instantiation for variables M and evidence e. Its main body sets m to some instantiation of M and p to the probability of m, e, both global, then calls $\text{BB}_{\text{MAP AUX}}(e, M)$ and returns m. The recursion $\text{BB}_{\text{MAP AUX}}(i, \mathbf{X})$ is:

- Line 1: if $\mathbf{X}$ is empty then (leaf node)
- Line 2: if $\Pr(i) > p$, then $m \leftarrow i$ and $p \leftarrow \Pr(i)$
- Line 3: else if $\text{MAP}^u_P(\mathbf{X}, i) > p$ then
- Line 4: $X \leftarrow$ a variable in $\mathbf{X}$
- Line 5: for each value x of variable X do
- Line 6: $\text{BB}_{\text{MAP AUX}}(ix, \mathbf{X} \setminus \{X\})$
- Line 7: end for
- Line 8: end if

Consider now the following heuristics:

- On Line 4, choose variable X that maximizes M_X / T_X, where

$$M_X = \max_x B_x, \qquad T_X = \sum_{B_x \geq p} B_x,$$

and p is the probability on Line 2 of Algorithm 31.

- On Line 5, choose values x in decreasing order of B_x.

Implement a version of Algorithm 31 that employs these heuristics and compare its performance with and without the heuristics. Note: The jointree algorithm can be used to compute the bounds B_x efficiently (see Exercise 10.20).

Solution. B_x is the pruning test that the child $(ix, \mathbf{X} \setminus \{X\})$ would run on its own Line 3, so the table $\{B_x\}$ is a one-step lookahead over every branch. By Exercise 10.20, one inward-outward jointree propagation with evidence i, MAP variables maximized out of messages and all others summed out, yields

$$B_x \;=\; \max_{(C_j \cap M) \setminus \{X\}} \sum_{C_j \setminus M} f_j(C_j)\Big|_{X=x}$$

for every remaining MAP variable $X \in C_j$ and every value x at once, at the $O(n \exp(w))$ cost of the single bound the unmodified algorithm computes – so the heuristics are informationally rich but computationally free. Two things then come at no charge: $\mathrm{MAP}_P^u(\mathbf{X}, i) = \max_x B_x = M_X$ for any $X \in \mathbf{X}$, which answers Line 3; and a child with $B_x \leq p$ may be skipped without a call, since it would return at its own Line 3 (at a leaf child no MAP variable is left, so $B_x = \Pr(ix)$ and the skip agrees with Line 2).

Lines 3 to 8 become: (1) one propagation, extracting $\{B_x\}$; (2) return if $\max_x B_x \leq p$; (3) let $X^\star$ maximize M_X/T_X, ties toward smaller T_X; (4) visit the values of $X^\star$ in decreasing B_x, breaking as soon as $B_x \leq p$ – re-tested inside the loop, since p may have grown since step (1) and never falls.

Since M_X is nearly the node's bound and so nearly independent of X, maximizing M_X/T_X is minimizing the bound mass of the children that survive the incumbent p: most-constrained-variable-first, measured in bound mass rather than domain size. Decreasing B_x is best-first ordering inside the depth-first search, installing a strong incumbent before any sibling is touched.

Measured on random networks – n binary variables, at most three parents per node, CPT rows from a symmetric Dirichlet with concentration 0.5, a random M of size m, three further variables as evidence, one fixed unconstrained order per instance, the same seed instantiation for every variant, all variants returning the same MAP probability – counting $\mathrm{BB}_{\mathrm{MAP\,AUX}}$ nodes, which is also the number of propagations:

plain is Algorithm 31 verbatim; *sibling* adds only the free pruning of children with $B_x \leq p$; *var* and *val* add the variable and value heuristics on top of *sibling*; *both* uses all of it.

variant	total nodes, n=16, m=8, 20 instances	median	total nodes, n=18, m=10, 12 instances	median
plain	1118	59	1240	109
sibling	701	34	780	70
var	896	39	1038	89
val	197	9	155	11
both	181	9	139	11

So *both* is 6.2 times cheaper than plain on the smaller instances and 8.9 times cheaper on the larger ones, the advantage growing with m. The medians are the sharper figure: a single root-to-leaf descent visits $m + 1$ nodes, which is exactly the heuristic median in both settings, so on a typical instance the value heuristic descends straight to an optimal leaf and the incumbent prunes every sibling with no backtracking.

Almost all of the gain is the value ordering; the M_X/T_X choice applied alone *hurt* (896 against 701, 1038 against 780), recovering only about ten per cent when combined. The standalone loss is the criterion's own definition: minimizing T_X prefers variables whose surviving child has a small bound, and such a child would be pruned soon anyway, so effort goes to the least informative decisions first. Node count is a fair runtime proxy only because every variant pays one propagation per node – computing each B_x by a separate $\mathrm{VE}_{\mathrm{MAP}}$ call would cost a factor $2|\mathbf{X}|$ per node and turn the saving into a loss.

9.4 Exercises 10.22–10.22

Problem 10.22. We can modify Algorithm 13, RC1, from Chapter 8 to answer MPE queries by replacing summation with maximization, leading to RC MPE shown in Algorithm 33. This algorithm computes the MPE probability but not the MPE instantiations.
Recall the setting of Chapter 8. A dtree T for a Bayesian network is a full binary tree whose leaves correspond to the network CPTs (Definition 8.2); T^l and T^r denote the left and right children of node T, and $\text{vars}(T)$ denotes the variables appearing at the leaves of T. The cutset of a nonleaf node is

$$\text{cutset}(T) \;=\; \big(\text{vars}(T^l) \cap \text{vars}(T^r)\big) \setminus \text{acutset}(T),$$

where $\text{acutset}(T)$ is the union of the cutsets of all ancestors of T (Definition 8.3). Algorithm 33 is then as follows; it is called initially on the dtree root with the given evidence $\mathbf{e}$, and it returns the MPE probability $\text{MPE}_P(\mathbf{e}) = \max_{\mathbf{x}\sim\mathbf{e}} \Pr(\mathbf{x})$.

RC MPE(T, $\mathbf{e}$)

```
1: if T is a leaf node then
2:     Θ_{X|U} ← CPT associated with node T
3:     u ← instantiation of parents U consistent with evidence e
4:     if X has value x in evidence e then
5:         return θ_{x|u}
6:     else
7:         return max_x θ_{x|u}
8:     end if
9: else
10:    p ← 0
11:    C ← cutset(T)
12:    for each instantiation c consistent with evidence e, c ~ e, do
13:        p ← max(p, RC MPE(T^l, ec) · RC MPE(T^r, ec))
14:    end for
15:    return p
16: end if
```

(a) How can we modify Algorithm RC MPE so that it returns the number of MPE instantiations?
(b) How can we modify Algorithm RC MPE so that it returns an NNF circuit that encodes the set of MPE instantiations, that is, the circuit models are precisely the MPE instantiations?

Solution. (a) Have every call return a pair $(p, \#)$, where $\#$ counts the maximizers.
Write $\text{free}(T, \mathbf{e}) = \text{vars}(T) \setminus \text{vars}(\mathbf{e})$, let f_T be the product of the CPTs at the leaves of T, and let $\text{Sol}(T, \mathbf{e})$ be the set of instantiations $\mathbf{y}$ of $\text{free}(T, \mathbf{e})$ attaining $\text{RC MPE}(T, \mathbf{e}) = \max_{\mathbf{y}} f_T(\mathbf{ey})$, which at the root is the set of MPE instantiations for the original evidence $\mathbf{e}_0$. Here $\mathbf{e}$ is $\mathbf{e}_0$ together with the cutsets of all ancestors of T, so $\text{acutset}(T) \subseteq \text{vars}(\mathbf{e})$. Two facts from Definition 8.3 drive both parts.
(F1) At a leaf with CPT $\Theta_{X|\mathbf{U}}$ every parent $U \in \mathbf{U}$ is already instantiated, which is what makes Line 3 well defined: U owns a CPT at some other leaf T', so $U \in \text{vars}(A^l) \cap \text{vars}(A^r)$ for the deepest common ancestor A of T and T', whence $U \in \text{cutset}(A) \cup \text{acutset}(A) \subseteq \text{acutset}(T)$. Hence $\text{free}(T, \mathbf{e})$ is $\{X\}$ or $\emptyset$.
(F2) At a nonleaf with $\mathbf{C} = \text{cutset}(T)$ and $\mathbf{C}'$ its part not fixed by $\mathbf{e}$, for every $\mathbf{c} \sim \mathbf{e}$

$$\text{free}(T, \mathbf{e}) \;=\; \mathbf{C}' \;\cup\; \text{free}(T^l, \mathbf{ec}) \;\cup\; \text{free}(T^r, \mathbf{ec}),$$

pairwise disjointly, since $\text{vars}(T^l) \cap \text{vars}(T^r) \subseteq \mathbf{C} \cup \text{acutset}(T)$ is instantiated in $\mathbf{ec}$; and the two subtree sets depend on $\mathbf{c}$ only through $\text{vars}(\mathbf{e}) \cup \mathbf{C}$, hence not at all. With $f_T = f_{T^l} f_{T^r}$, this is what licenses Line 13.
The three returns become:

- Line 5: $(\theta_{x|\mathbf{u}},\ 1)$, since $\text{free}(T, \mathbf{e}) = \emptyset$ by (F1).
- Line 7: $(p,\ |\{x : \theta_{x|\mathbf{u}} = p\}|)$ with $p = \max_x \theta_{x|\mathbf{u}}$.
- Lines 10-15: with $p_{\mathbf{c}} = p_l p_r$ and $\#_{\mathbf{c}} = \#_l \#_r$, return

$$\Big(p,\ \sum_{\mathbf{c}\,:\,p_{\mathbf{c}}=p} \#_{\mathbf{c}}\Big), \qquad p=\max_{\mathbf{c}} p_{\mathbf{c}},$$

in one sweep of the loop: reset the count on a strict improvement, add on a tie.

- Whenever a call obtains $p=0$, discard that count and return instead $\big(0,\ \prod_{X\in\mathrm{free}(T,\mathbf{e})}|X|\big)$, where $|X|$ is the number of values of X.

Correctness is an induction on the dtree, the leaves being (F1). At a nonleaf, (F2) factors every $\mathbf{y}$ uniquely as $\mathbf{c}'\mathbf{y}_l\mathbf{y}_r$, and

$$f_T(\mathbf{e}\mathbf{y}) = f_{T^l}(\mathbf{e}\mathbf{c}\mathbf{y}_l)\, f_{T^r}(\mathbf{e}\mathbf{c}\mathbf{y}_r) \ \le\ p_l p_r = p_{\mathbf{c}} \ \le\ p,$$

which is the correctness of RC MPE itself. If $p>0$, equality forces $\mathbf{y}_l\in\mathrm{Sol}(T^l,\mathbf{e}\mathbf{c})$ and $\mathbf{y}_r\in\mathrm{Sol}(T^r,\mathbf{e}\mathbf{c})$: writing $a\le p_l$ and $b\le p_r$ for the two factors, $ab=p_lp_r>0$ gives $b>0$, and $a<p_l$ would give $ab<p_lb\le p_lp_r$. Hence

$$\begin{aligned}\mathrm{Sol}(T,\mathbf{e}) \ &=\ \bigcup_{\mathbf{c}\,:\,p_{\mathbf{c}}=p} S_{\mathbf{c}},\\ S_{\mathbf{c}} \ &=\ \big\{\mathbf{c}'\mathbf{y}_l\mathbf{y}_r \ :\ \mathbf{y}_l\in\mathrm{Sol}(T^l,\mathbf{e}\mathbf{c}),\ \mathbf{y}_r\in\mathrm{Sol}(T^r,\mathbf{e}\mathbf{c})\big\},\end{aligned}$$

a disjoint union because distinct $\mathbf{c}\sim\mathbf{e}$ disagree on $\mathbf{C}'$, and cardinalities give the recurrence. The $p=0$ line is not optional: the product rule undercounts there (a subtree with $p_l=0$ beside one with a unique maximizer), whereas in fact every $\mathbf{y}$ has $f_T(\mathbf{e}\mathbf{y})=0$, so $\mathrm{Sol}(T,\mathbf{e})$ is all of $\mathrm{free}(T,\mathbf{e})$ – a set read off the dtree at no search cost. Since a node with $p>0$ has $p_l,p_r>0$ for every surviving $\mathbf{c}$, the two clauses together are exact.

The extra work is a constant per cutset instantiation, so the bounds are unchanged: $O(n\exp(w\log n))$ time and $O(wn)$ space by Theorem 8.2, or, caching the pair $(p,\#)$ in RC2 (Algorithm 14) under context(T) (Definition 8.5), $O(n\exp(w))$ time and space by Theorem 8.4 – caching stays sound because f_T mentions only vars(T), so $\#$ depends on $\mathbf{e}$ exactly as p does.

(b) Have every call return (p,Γ) instead, replacing "add counts" by an or-node and "multiply counts" by an and-node. Use the indicator variables λ_x of Chapter 11 and

$$\mathrm{term}(\mathbf{z}) \ =\ \bigwedge_{X\in\mathbf{Z}}\Big(\lambda_x\wedge\bigwedge_{x'\neq x}\neg\lambda_{x'}\Big), \qquad x \text{ the value of } X \text{ in } \mathbf{z},$$

so that network instantiations correspond one to one with models over the indicators.

- Line 5: $(\theta_{x|\mathbf{u}},\top)$, the empty and-node; the literal for X is contributed by the root or by the ancestor whose cutset contains X.
- Line 7: $\Gamma=\bigvee_{x\,:\,\theta_{x|\mathbf{u}}=p}\mathrm{term}(x)$.
- Lines 10-15: $\Gamma_{\mathbf{c}}=\mathrm{term}(\mathbf{c}')\wedge\Gamma_l\wedge\Gamma_r$ and $\Gamma=\bigvee_{\mathbf{c}\,:\,p_{\mathbf{c}}=p}\Gamma_{\mathbf{c}}$.
- Zero case: $\Gamma=\bigwedge_{X\in\mathrm{free}(T,\mathbf{e})}\bigvee_x\mathrm{term}(x)$.
- At the root, conjoin $\mathrm{term}(\mathbf{e}_0)$ if the models are to be full network instantiations.

That the models of Γ are $\mathrm{Sol}(T,\mathbf{e})$ is the displayed set equation of part (a) read as a recursive definition of a circuit, the or-node realizing the disjoint union and the and-node the Cartesian product over disjoint variable sets.

Γ is moreover a smooth d-DNNF (Section 2.7). Decomposable: the children of $\Gamma_{\mathbf{c}}$ mention the indicators of $\mathbf{C}'$, $\mathrm{free}(T^l,\mathbf{e}\mathbf{c})$ and $\mathrm{free}(T^r,\mathbf{e}\mathbf{c})$, pairwise disjoint by (F2). Deterministic: two disjuncts of a leaf or-node contain λ_x and $\neg\lambda_x$, and two disjuncts $\Gamma_{\mathbf{c}},\Gamma_{\mathbf{d}}$ disagree on some variable of $\mathbf{C}'$. Smooth: every disjunct mentions the indicators of $\mathrm{free}(T,\mathbf{e})$, the same set for every $\mathbf{c}$ by the last clause of (F2). So the MPE instantiations enumerate from Γ in time linear in their number and $|\Gamma|$ (Exercise 2.13), and the standard smooth-d-DNNF model count recovers part (a) arithmetic for arithmetic.

Writing $\mathbf{C}^{\#}$ for the number of instantiations of $\mathbf{C}$, each call adds one or-node, at most $\mathbf{C}^{\#}$ and-nodes, and one $\mathrm{term}(\mathbf{c}')$ per and-node; since $\sum_{X\in\mathbf{C}}|X|\le\prod_{X\in\mathbf{C}}|X|=\mathbf{C}^{\#}$, the extra factor is absorbed and $|\Gamma|=O(n\exp(w\log n))$. Caching (p,Γ) in RC2 shares nodes, so Γ becomes a DAG of size $O(n\exp(w))$ by Theorem 8.4: the entire set of MPE instantiations costs no more than the MPE probability alone.

10 The Complexity of Probabilistic Inference

Contents

10.1 Exercises 11.1–11.7 162
10.2 Exercises 11.8–11.10 170

10.1 Exercises 11.1–11.7

Problem 11.1. Prove Theorem 11.1.
Recall the construction of Section 11.3. Given a propositional sentence α over variables $X_1, \ldots, X_n$, the Bayesian network N_α has a single leaf node S_α and is built inductively over the structure of α:

- If α is a single variable X, its network has the single binary root node $S_\alpha = X$ with CPT $\theta_x = 1/2$ and $\theta_{\bar{x}} = 1/2$.
- If α is $\neg\beta$, the network for α is obtained from the network for β by adding node S_α with parent S_β and CPT (redundant rows omitted)

S_β	S_α	$\theta_{S_\alpha \mid S_\beta}$
true	true	0
false	true	1

- If α is $\beta \vee \gamma$, the network is obtained from the networks for β and γ by adding S_α with parents S_β, S_γ and CPT

S_β	S_γ	S_α	$\theta_{S_\alpha \mid S_\beta, S_\gamma}$
true	true	true	1
true	false	true	1
false	true	true	1
false	false	true	0

- If α is $\beta \wedge \gamma$, the network is obtained analogously with CPT

S_β	S_γ	S_α	$\theta_{S_\alpha \mid S_\beta, S_\gamma}$
true	true	true	1
true	false	true	0
false	true	true	0
false	false	true	0

The networks for subsentences share the root node of a propositional variable whenever that variable occurs more than once. For instance, for $\alpha : (X_1 \vee X_2 \vee \neg X_3) \wedge ((X_3 \wedge X_4) \vee \neg X_5)$ the network of Figure 11.1 has roots $X_1, \ldots, X_5$; a $\neg$ node with parent X_3; a $\vee$ node with parents X_1, X_2 and that $\neg$ node; a $\wedge$ node with parents X_3, X_4; a $\neg$ node with parent X_5; a second $\vee$ node with parents the $\wedge$ node and the second $\neg$ node; and the leaf S_α, a $\wedge$ node whose parents are the two $\vee$ nodes.
Theorem 11.1 states that the distribution Pr induced by N_α satisfies

$$Pr(x_1, \ldots, x_n, S_\alpha = \text{true}) = \begin{cases} 0, & \text{if } x_1, \ldots, x_n \models \neg\alpha \\ 1/2^n, & \text{if } x_1, \ldots, x_n \models \alpha. \end{cases}$$

Solution. Every internal node is a logic gate. Inspecting the three CPTs, the $\neg$ node gives parameter 1 to the value $\neg S_\gamma$, the $\vee$ node to $S_{\gamma_1} \vee S_{\gamma_2}$, and the $\wedge$ node to $S_{\gamma_1} \wedge S_{\gamma_2}$, and 0 to the other value.
Fix an instantiation $x_1, \ldots, x_n$ of $\mathbf{X} = \{X_1, \ldots, X_n\}$, read also as a truth assignment, and let $\beta_1, \ldots, \beta_m$ be the non-atomic subsentences for which the construction created a node, $\beta_m = \alpha$. Define $\mathbf{s}$ by

$$S_{\beta_j} = \text{true} \quad \text{iff} \quad x_1, \ldots, x_n \models \beta_j.$$

Then $\mathbf{s}$ is exactly the instantiation the gates force, by induction on subsentence structure: the atom X_i carries x_i; and the $\neg$, $\vee$, $\wedge$ steps are the gate identities above, using for $\neg$ that a truth assignment is complete, so $x_1, \ldots, x_n \not\models \gamma$ iff $x_1, \ldots, x_n \models \neg\gamma$.
By the chain rule for Bayesian networks (Chapter 3), for any instantiation $\mathbf{s}'$ of the internal nodes,

$$Pr(x_1, \ldots, x_n, \mathbf{s}') = \prod_{i=1}^{n} \theta_{x_i} \cdot \prod_{j=1}^{m} \theta_{s'_{\beta_j} \mid \mathbf{p}'_j} = \frac{1}{2^n} \prod_{j=1}^{m} \theta_{s'_{\beta_j} \mid \mathbf{p}'_j},$$

with $\mathbf{p}'_j$ the parent values under $x_1, \ldots, x_n, \mathbf{s}'$ and each root contributing $\theta_{x_i} = 1/2$. For $\mathbf{s}' = \mathbf{s}$ every remaining factor is 1, so $Pr(x_1, \ldots, x_n, \mathbf{s}) = 1/2^n$. For $\mathbf{s}' \neq \mathbf{s}$, let S_{β_j} be topmost in the DAG order among the nodes of disagreement (the disagreement set is nonempty and the DAG order well founded); its parents then carry their $\mathbf{s}$-values, so its factor is 0 and the product vanishes.

Summing over the internal nodes with S_α held true therefore leaves at most the one term $\mathbf{s}$, and that term occurs iff $s_\alpha = \text{true}$, that is iff $x_1, \ldots, x_n \models \alpha$. Hence

$$Pr(x_1, \ldots, x_n, S_\alpha = \text{true}) = \begin{cases} 1/2^n, & \text{if } x_1, \ldots, x_n \models \alpha \\ 0, & \text{if } x_1, \ldots, x_n \models \neg\alpha, \end{cases}$$

which is (11.2). ■

Problem 11.2. Prove Theorem 11.2.
Theorem 11.2 (Reducing SAT to D-MPE): let α be a propositional sentence over variables $X_1, \ldots, X_n$ and let N_α be the Bayesian network constructed for it in Section 11.3 (roots $X_1, \ldots, X_n$ with uniform CPTs, one deterministic gate node S_β per non-atomic subsentence β, leaf S_α; see Exercise 11.1 for the construction and its CPTs). Then there is a variable instantiation $x_1, \ldots, x_n$ that satisfies sentence α if and only if there is a variable instantiation $\mathbf{y}$ of network N_α such that $Pr(\mathbf{y}, S_\alpha = \text{true}) > 0$. (Here $\mathbf{y}$ instantiates every variable of N_α except the leaf S_α, so that $\mathbf{y}, S_\alpha = \text{true}$ is a complete network instantiation.)

Solution. Both directions are Exercise 11.1, with $\mathbf{y}$ ranging over instantiations of the roots $\mathbf{X}$ and the gate nodes $S_{\beta_1}, \ldots, S_{\beta_{m-1}}$.
(i) If $x_1, \ldots, x_n \models \alpha$, let $\mathbf{s}$ be the gate instantiation it induces (S_{β_j} true iff $x_1, \ldots, x_n \models \beta_j$) and take $\mathbf{y}$ to be $x_1, \ldots, x_n$ together with the values $\mathbf{s}$ gives $S_{\beta_1}, \ldots, S_{\beta_{m-1}}$. Since $x_1, \ldots, x_n \models \alpha$ we have $s_\alpha = \text{true}$, so $\mathbf{y}, S_\alpha = \text{true}$ is the complete instantiation $x_1, \ldots, x_n, \mathbf{s}$ and

$$Pr(\mathbf{y}, S_\alpha = \text{true}) = \frac{1}{2^n} > 0.$$

(ii) Conversely, suppose $Pr(\mathbf{y}, S_\alpha = \text{true}) > 0$ and let $x_1, \ldots, x_n$ be the restriction of $\mathbf{y}$ to $\mathbf{X}$. Marginalizing the gate nodes other than S_α only increases the number, so

$$Pr(x_1, \ldots, x_n, S_\alpha = \text{true}) \;\geq\; Pr(\mathbf{y}, S_\alpha = \text{true}) \;>\; 0,$$

and Theorem 11.1 makes the left side 0 whenever $x_1, \ldots, x_n \models \neg\alpha$. Hence $x_1, \ldots, x_n \models \alpha$ and α is satisfiable. ■
The reduction is linear time: N_α has at most $|\alpha| + n$ nodes, each gate has at most two parents, and each parameter is 0, 1, or 1/2.

Problem 11.3. Consider the CPT in Table 11.4. Generate a CNF encoding for this CPT according to (11.3) and (11.4). Show the weights of all variables in the encoding.
Table 11.4 is the CPT of a variable C with values c_1, c_2, c_3 and parents A (values a_1, a_2) and B (values b_1, b_2):

A	B	C	$\theta_{C\|A,B}$
a_1	b_1	c_1	.2
a_1	b_1	c_2	.1
a_1	b_1	c_3	.7
a_1	b_2	c_1	0
a_1	b_2	c_2	0
a_1	b_2	c_3	1
a_2	b_1	c_1	.5
a_2	b_1	c_2	.2
a_2	b_1	c_3	.3
a_2	b_2	c_1	.2
a_2	b_2	c_2	0
a_2	b_2	c_3	.8

Recall the first encoding. For each network variable X with parents $\mathbf{U}$ there is an indicator variable I_x for each value x of X and a parameter variable $P_{x|\mathbf{u}}$ for each family instantiation $x\mathbf{u}$. The indicator clauses for a variable X with values $x_1, \ldots, x_k$ are

$$I_{x_1} \vee I_{x_2} \vee \ldots \vee I_{x_k}, \qquad \neg I_{x_i} \vee \neg I_{x_j} \ \text{ for } i < j, \tag{11.3}$$

and the parameter clauses are, for each parameter $\theta_{x|u_1,\ldots,u_m}$,

$$I_{u_1} \wedge I_{u_2} \wedge \ldots \wedge I_{u_m} \wedge I_x \iff P_{x|u_1,\ldots,u_m}. \tag{11.4}$$

Solution. Seven indicator variables,

$$I_{a_1}, I_{a_2}, \qquad I_{b_1}, I_{b_2}, \qquad I_{c_1}, I_{c_2}, I_{c_3},$$

and twelve parameter variables, one per row of Table 11.4,

$$\begin{array}{lll} P_{c_1|a_1b_1}, & P_{c_2|a_1b_1}, & P_{c_3|a_1b_1}, \\ P_{c_1|a_1b_2}, & P_{c_2|a_1b_2}, & P_{c_3|a_1b_2}, \\ P_{c_1|a_2b_1}, & P_{c_2|a_2b_1}, & P_{c_3|a_2b_1}, \\ P_{c_1|a_2b_2}, & P_{c_2|a_2b_2}, & P_{c_3|a_2b_2}. \end{array}$$

Indicator clauses, by (11.3), eight in all since A, B are binary and C ternary:

$$\begin{array}{ll} A: & I_{a_1} \vee I_{a_2}, \qquad \neg I_{a_1} \vee \neg I_{a_2} \\ B: & I_{b_1} \vee I_{b_2}, \qquad \neg I_{b_1} \vee \neg I_{b_2} \\ C: & I_{c_1} \vee I_{c_2} \vee I_{c_3}, \\ & \neg I_{c_1} \vee \neg I_{c_2}, \quad \neg I_{c_1} \vee \neg I_{c_3}, \quad \neg I_{c_2} \vee \neg I_{c_3}. \end{array}$$

Parameter clauses, by (11.4), one equivalence per row:

$$\begin{array}{ll} I_{a_1} \wedge I_{b_1} \wedge I_{c_1} & \iff P_{c_1|a_1b_1} \\ I_{a_1} \wedge I_{b_1} \wedge I_{c_2} & \iff P_{c_2|a_1b_1} \\ I_{a_1} \wedge I_{b_1} \wedge I_{c_3} & \iff P_{c_3|a_1b_1} \\ I_{a_1} \wedge I_{b_2} \wedge I_{c_1} & \iff P_{c_1|a_1b_2} \\ I_{a_1} \wedge I_{b_2} \wedge I_{c_2} & \iff P_{c_2|a_1b_2} \\ I_{a_1} \wedge I_{b_2} \wedge I_{c_3} & \iff P_{c_3|a_1b_2} \\ I_{a_2} \wedge I_{b_1} \wedge I_{c_1} & \iff P_{c_1|a_2b_1} \\ I_{a_2} \wedge I_{b_1} \wedge I_{c_2} & \iff P_{c_2|a_2b_1} \\ I_{a_2} \wedge I_{b_1} \wedge I_{c_3} & \iff P_{c_3|a_2b_1} \\ I_{a_2} \wedge I_{b_2} \wedge I_{c_1} & \iff P_{c_1|a_2b_2} \\ I_{a_2} \wedge I_{b_2} \wedge I_{c_2} & \iff P_{c_2|a_2b_2} \\ I_{a_2} \wedge I_{b_2} \wedge I_{c_3} & \iff P_{c_3|a_2b_2} \end{array}$$

Each equivalence is four CNF clauses, the IP clause

$$\neg I_{a_i} \vee \neg I_{b_j} \vee \neg I_{c_k} \vee P_{c_k|a_ib_j}$$

and the three PI clauses

$$\neg P_{c_k|a_ib_j} \vee I_{a_i}, \quad \neg P_{c_k|a_ib_j} \vee I_{b_j}, \quad \neg P_{c_k|a_ib_j} \vee I_{c_k},$$

so the encoding has $12 \times 4 + 8 = 56$ clauses over 19 Boolean variables.
Weights, by Theorem 11.7: $Wt(I_x) = Wt(\neg I_x) = Wt(\neg P_{x|\mathbf{u}}) = 1$ and $Wt(P_{x|\mathbf{u}}) = \theta_{x|\mathbf{u}}$, so that the family contributes $\theta_{c_k|a_ib_j}$ to a model selecting $a_ib_jc_k$, exactly the chain-rule factor.

variable	$Wt(\cdot)$	$Wt(\neg\cdot)$
I_{a_1}	1	1
I_{a_2}	1	1
I_{b_1}	1	1
I_{b_2}	1	1
I_{c_1}	1	1
I_{c_2}	1	1
I_{c_3}	1	1
$P_{c_1\|a_1b_1}$	.2	1
$P_{c_2\|a_1b_1}$	.1	1
$P_{c_3\|a_1b_1}$	.7	1
$P_{c_1\|a_1b_2}$	0	1
$P_{c_2\|a_1b_2}$	0	1
$P_{c_3\|a_1b_2}$	1	1
$P_{c_1\|a_2b_1}$	.5	1
$P_{c_2\|a_2b_1}$	.2	1
$P_{c_3\|a_2b_1}$	.3	1
$P_{c_1\|a_2b_2}$	.2	1
$P_{c_2\|a_2b_2}$	0	1
$P_{c_3\|a_2b_2}$	.8	1

Problem 11.4. Consider the CPT in Table 11.4. Generate a CNF encoding for this CPT according to (11.5) and (11.6). Show the weights of all variables in the encoding.

Table 11.4 is the CPT of a variable C with values c_1, c_2, c_3 and parents A (values a_1, a_2) and B (values b_1, b_2):

A	B	C	$\theta_{C\|A,B}$
a_1	b_1	c_1	.2
a_1	b_1	c_2	.1
a_1	b_1	c_3	.7
a_1	b_2	c_1	0
a_1	b_2	c_2	0
a_1	b_2	c_3	1
a_2	b_1	c_1	.5
a_2	b_1	c_2	.2
a_2	b_1	c_3	.3
a_2	b_2	c_1	.2
a_2	b_2	c_2	0
a_2	b_2	c_3	.8

Recall the second encoding. It assumes an ordering of the values of each variable and uses, for each network variable X with parents $\mathbf{U}$, an indicator variable I_x for each value x of X, and a parameter variable $Q_{x|\mathbf{u}}$ for each family instantiation $x\mathbf{u}$ in which x is not the last value in the ordering of X. The indicator clauses are as in the first encoding,

$$I_{x_1} \vee I_{x_2} \vee \ldots \vee I_{x_k}, \qquad \neg I_{x_i} \vee \neg I_{x_j} \ \text{ for } i < j, \tag{11.5}$$

and, for $x_1 < x_2 < \cdots < x_k$ the value order of X and $\mathbf{u} = u_1, \ldots, u_m$ an instantiation of $\mathbf{U}$, the parameter clauses are

$$\begin{aligned} I_{u_1} \wedge \ldots \wedge I_{u_m} \wedge \neg Q_{x_1|\mathbf{u}} \wedge \ldots \wedge \neg Q_{x_{i-1}|\mathbf{u}} \wedge Q_{x_i|\mathbf{u}} &\Longrightarrow I_{x_i}, \quad i < k \\ I_{u_1} \wedge \ldots \wedge I_{u_m} \wedge \neg Q_{x_1|\mathbf{u}} \wedge \ldots \wedge \neg Q_{x_{k-1}|\mathbf{u}} &\Longrightarrow I_{x_k}. \end{aligned} \tag{11.6}$$

Assume the value orders $a_1 < a_2$, $b_1 < b_2$, and $c_1 < c_2 < c_3$.

Solution. The seven indicators are as in Exercise 11.3,

$$I_{a_1},\, I_{a_2}, \qquad I_{b_1},\, I_{b_2}, \qquad I_{c_1},\, I_{c_2},\, I_{c_3},$$

but since c_3 is last in the value order of C, only c_1 and c_2 generate parameter variables, eight rather than twelve:

$$\begin{aligned} &Q_{c_1|a_1b_1},\; Q_{c_2|a_1b_1}, \qquad Q_{c_1|a_1b_2},\; Q_{c_2|a_1b_2}, \\ &Q_{c_1|a_2b_1},\; Q_{c_2|a_2b_1}, \qquad Q_{c_1|a_2b_2},\; Q_{c_2|a_2b_2}. \end{aligned}$$

Indicator clauses, by (11.5), exactly as in Exercise 11.3:

$$\begin{aligned} A:&\quad I_{a_1} \vee I_{a_2}, \qquad \neg I_{a_1} \vee \neg I_{a_2} \\ B:&\quad I_{b_1} \vee I_{b_2}, \qquad \neg I_{b_1} \vee \neg I_{b_2} \\ C:&\quad I_{c_1} \vee I_{c_2} \vee I_{c_3}, \\ &\quad \neg I_{c_1} \vee \neg I_{c_2}, \quad \neg I_{c_1} \vee \neg I_{c_3}, \quad \neg I_{c_2} \vee \neg I_{c_3}. \end{aligned}$$

Parameter clauses, by (11.6): three per parent instantiation, twelve in all. For $\mathbf{u} = a_i b_j$,

$$\begin{aligned} I_{a_i} \wedge I_{b_j} \wedge Q_{c_1|\mathbf{u}} &\Longrightarrow I_{c_1} \\ I_{a_i} \wedge I_{b_j} \wedge \neg Q_{c_1|\mathbf{u}} \wedge Q_{c_2|\mathbf{u}} &\Longrightarrow I_{c_2} \\ I_{a_i} \wedge I_{b_j} \wedge \neg Q_{c_1|\mathbf{u}} \wedge \neg Q_{c_2|\mathbf{u}} &\Longrightarrow I_{c_3} \end{aligned}$$

taken over $\mathbf{u} = a_1b_1$, a_1b_2, a_2b_1, a_2b_2, or in clausal form

$$\begin{aligned} &\neg I_{a_i} \vee \neg I_{b_j} \vee \neg Q_{c_1|\mathbf{u}} \vee I_{c_1} \\ &\neg I_{a_i} \vee \neg I_{b_j} \vee Q_{c_1|\mathbf{u}} \vee \neg Q_{c_2|\mathbf{u}} \vee I_{c_2} \\ &\neg I_{a_i} \vee \neg I_{b_j} \vee Q_{c_1|\mathbf{u}} \vee Q_{c_2|\mathbf{u}} \vee I_{c_3}. \end{aligned}$$

Weights, by Theorem 11.8: indicator literals all 1, and

$$Wt(Q_{x|\mathbf{u}}) = \frac{\theta_{x|\mathbf{u}}}{1 - \sum_{x'<x} \theta_{x'|\mathbf{u}}}, \qquad Wt(\neg Q_{x|\mathbf{u}}) = 1 - Wt(Q_{x|\mathbf{u}}),$$

the conditional probability that $C = x$ given that C is none of the earlier values; so $Wt(Q_{c_1|a_1b_1}) = .2/1 = .2$ and $Wt(Q_{c_2|a_1b_1}) = .1/.8 = .125$, and likewise elsewhere (Check!).

variable	$Wt(\cdot)$	$Wt(\neg\cdot)$
I_{a_1}	1	1
I_{a_2}	1	1
I_{b_1}	1	1
I_{b_2}	1	1
I_{c_1}	1	1
I_{c_2}	1	1
I_{c_3}	1	1
$Q_{c_1\|a_1b_1}$	.2	.8
$Q_{c_2\|a_1b_1}$	.125	.875
$Q_{c_1\|a_1b_2}$	0	1
$Q_{c_2\|a_1b_2}$	0	1
$Q_{c_1\|a_2b_1}$	.5	.5
$Q_{c_2\|a_2b_1}$	.4	.6
$Q_{c_1\|a_2b_2}$	.2	.8
$Q_{c_2\|a_2b_2}$	0	1

The branch weights q_1, $(1-q_1)q_2$, $(1-q_1)(1-q_2)$ reproduce each row of Table 11.4, as they must:

$\mathbf{u}$	c_1	c_2	c_3
a_1b_1	.2	$.8 \times .125 = .1$	$.8 \times .875 = .7$
a_1b_2	0	$1 \times 0 = 0$	$1 \times 1 = 1$
a_2b_1	.5	$.5 \times .4 = .2$	$.5 \times .6 = .3$
a_2b_2	.2	$.8 \times 0 = 0$	$.8 \times 1 = .8$

Problem 11.5. Consider Theorems 11.7 and 11.8. Provide a different weight function $Wt(.)$ that depends on evidence e for which the following would hold: $Pr(e) = \text{WMC}(\Delta_N)$.
Both theorems accommodate evidence $e = e_1, \ldots, e_k$ by conjoining the indicator literals of the evidence to the CNF encoding Δ_N of the network,

$$Pr(e) = \text{WMC}(\Delta_N \wedge I_{e_1} \wedge \ldots \wedge I_{e_k}),$$

with the weights of Theorem 11.7,

$$Wt(I_x) = Wt(\neg I_x) = Wt(\neg P_{x|\mathbf{u}}) = 1, \qquad Wt(P_{x|\mathbf{u}}) = \theta_{x|\mathbf{u}},$$

or those of Theorem 11.8,

$$Wt(I_x) = Wt(\neg I_x) = 1, \quad Wt(Q_{x|\mathbf{u}}) = \frac{\theta_{x|\mathbf{u}}}{1 - \sum_{x' < x} \theta_{x'|\mathbf{u}}}, \quad Wt(\neg Q_{x|\mathbf{u}}) = 1 - Wt(Q_{x|\mathbf{u}}).$$

The task is to move the evidence out of the sentence and into the weights, so that the plain encoding Δ_N is left untouched.

Solution. Take Wt_e to agree with the weight function of the relevant theorem on every parameter literal and to set

$$Wt_e(I_x) = \begin{cases} 1, & \text{if } x \text{ is consistent with } e \\ 0, & \text{otherwise,} \end{cases} \qquad Wt_e(\neg I_x) = 1,$$

where a value x of X is consistent with e when e does not instantiate X, or instantiates it to x. All weights stay nonnegative, as Definition 11.3 requires, and only the positive indicators of values ruled out by e have changed.
First encoding. The models of Δ_N correspond one to one with network instantiations $\mathbf{x}$ (the correspondence underlying Theorem 11.7, Table 11.2): the clauses (11.3) force one true indicator per variable, and (11.4) then forces $P_{x|\mathbf{u}}$ true exactly for the family instantiations compatible with $\mathbf{x}$, every other literal occurring negatively with weight 1. So, by the chain rule,

$$Wt_e(\omega_{\mathbf{x}}) = \Big[\prod_{x \in \mathbf{x}} Wt_e(I_x)\Big] \cdot \prod_{\theta_{x|\mathbf{u}} \sim \mathbf{x}} \theta_{x|\mathbf{u}} = \Big[\prod_{x \in \mathbf{x}} Wt_e(I_x)\Big] \cdot Pr(\mathbf{x}),$$

and the bracket is 1 if $\mathbf{x} \sim e$ and 0 otherwise, so $\text{WMC}(\Delta_N) = \sum_{\mathbf{x} \sim e} Pr(\mathbf{x}) = Pr(e)$.
Second encoding. Here the models are not in one-to-one correspondence – the Q blocks of the unselected parent instantiations are unconstrained by (11.6) – so partition them by the instantiation $\mathbf{x}$ they select. Since Wt_e leaves every parameter weight unchanged, the argument for Theorem 11.8 gives each block parameter-weight total $Pr(\mathbf{x})$, the free Q variables contributing $Wt(Q) + Wt(\neg Q) = 1$ apiece. Every model of a block has the same indicator literals, hence the same extra factor $\prod_{x \in \mathbf{x}} Wt_e(I_x)$, again 1 iff $\mathbf{x} \sim e$. So $\text{WMC}(\Delta_N) = \sum_{\mathbf{x} \sim e} Pr(\mathbf{x}) = Pr(e)$. ■
Method (2): keep $Wt(I_x) = 1$ everywhere and instead set $Wt_e(\neg I_x) = 0$ for the values $x = e(X)$ asserted by the evidence. A model whose instantiation disagrees with e at X then contains $\neg I_{e(X)}$ and has weight 0, while one agreeing with e is unaffected, and the two computations above go through unchanged.

Problem 11.6. Consider the CPT in Table 11.4. Generate a W-MAXSAT encoding for this CPT.
Table 11.4 is the CPT of a variable C with values c_1, c_2, c_3 and parents A (values a_1, a_2) and B (values b_1, b_2):

A	B	C	$\theta_{C\mid A,B}$
a_1	b_1	c_1	.2
a_1	b_1	c_2	.1
a_1	b_1	c_3	.7
a_1	b_2	c_1	0
a_1	b_2	c_2	0
a_1	b_2	c_3	1
a_2	b_1	c_1	.5
a_2	b_1	c_2	.2
a_2	b_1	c_3	.3
a_2	b_2	c_1	.2
a_2	b_2	c_2	0
a_2	b_2	c_3	.8

Recall the reduction of MPE to W-MAXSAT of Section 11.7. It uses one indicator variable I_x per value x of each network variable X, and no parameter variables. For each variable X with values $x_1, \ldots, x_k$ it generates the hard indicator clauses

$$(I_{x_1} \vee I_{x_2} \vee \ldots \vee I_{x_k})^W \tag{11.8}$$

$$(\neg I_{x_i} \vee \neg I_{x_j})^W, \quad \text{for } i < j, \tag{11.9}$$

and for each network parameter $\theta_{x|u_1,\ldots,u_m}$ the single soft clause

$$(\neg I_x \vee \neg I_{u_1} \vee \ldots \vee \neg I_{u_m})^{-\log \theta_{x|u_1,\ldots,u_m}}, \tag{11.10}$$

where $-\log 0$ is defined to be the special hard weight W, and W is chosen larger than the sum of the weights of all soft clauses.

Solution. Only the seven indicators,

$$I_{a_1},\, I_{a_2}, \qquad I_{b_1},\, I_{b_2}, \qquad I_{c_1},\, I_{c_2},\, I_{c_3},$$

with no parameter variables at all. The hard indicator clauses, by (11.8) and (11.9), are

$$\begin{aligned}
A:&\quad (I_{a_1} \vee I_{a_2})^W, \qquad (\neg I_{a_1} \vee \neg I_{a_2})^W \\
B:&\quad (I_{b_1} \vee I_{b_2})^W, \qquad (\neg I_{b_1} \vee \neg I_{b_2})^W \\
C:&\quad (I_{c_1} \vee I_{c_2} \vee I_{c_3})^W, \\
&\quad (\neg I_{c_1} \vee \neg I_{c_2})^W, \quad (\neg I_{c_1} \vee \neg I_{c_3})^W, \quad (\neg I_{c_2} \vee \neg I_{c_3})^W,
\end{aligned}$$

and (11.10) gives one clause per row of Table 11.4:

$$\begin{array}{ll}
(\neg I_{c_1} \vee \neg I_{a_1} \vee \neg I_{b_1})^{-\log .2} & (\neg I_{c_2} \vee \neg I_{a_1} \vee \neg I_{b_1})^{-\log .1} \\
(\neg I_{c_3} \vee \neg I_{a_1} \vee \neg I_{b_1})^{-\log .7} & (\neg I_{c_1} \vee \neg I_{a_1} \vee \neg I_{b_2})^{W} \\
(\neg I_{c_2} \vee \neg I_{a_1} \vee \neg I_{b_2})^{W} & (\neg I_{c_3} \vee \neg I_{a_1} \vee \neg I_{b_2})^{-\log 1} \\
(\neg I_{c_1} \vee \neg I_{a_2} \vee \neg I_{b_1})^{-\log .5} & (\neg I_{c_2} \vee \neg I_{a_2} \vee \neg I_{b_1})^{-\log .2} \\
(\neg I_{c_3} \vee \neg I_{a_2} \vee \neg I_{b_1})^{-\log .3} & (\neg I_{c_1} \vee \neg I_{a_2} \vee \neg I_{b_2})^{-\log .2} \\
(\neg I_{c_2} \vee \neg I_{a_2} \vee \neg I_{b_2})^{W} & (\neg I_{c_3} \vee \neg I_{a_2} \vee \neg I_{b_2})^{-\log .8}
\end{array}$$

The three zero parameters give hard clauses, since $-\log 0 = W$; the unit parameter $\theta_{c_3|a_1b_2}$ gives weight $-\log 1 = 0$ and may be dropped. In natural logarithms (the base only rescales every weight by a common positive factor):

A	B	C	$\theta_{C\|A,B}$	clause weight $-\ln\theta$
a_1	b_1	c_1	.2	1.6094
a_1	b_1	c_2	.1	2.3026
a_1	b_1	c_3	.7	0.3567
a_1	b_2	c_1	0	W (hard)
a_1	b_2	c_2	0	W (hard)
a_1	b_2	c_3	1	0
a_2	b_1	c_1	.5	0.6931
a_2	b_1	c_2	.2	1.6094
a_2	b_1	c_3	.3	1.2040
a_2	b_2	c_1	.2	1.6094
a_2	b_2	c_2	0	W (hard)
a_2	b_2	c_3	.8	0.2231

The soft weights sum to 9.6078, so were this CPT the whole encoding any $W > 9.6078$, say $W = 10$, would serve; in a full network W must exceed the sum over all clauses.

A truth assignment violating an indicator clause pays at least W and cannot be optimal, so the optimal assignments are network instantiations; one selecting $a_i b_j c_k$ violates exactly the (11.10) clause of $\theta_{c_k|a_i b_j}$, all others having a true negative literal, so over the whole network

$$Pn(\hat{\omega}) = \sum_{\theta_{x|\mathbf{u}} \sim \mathbf{x}} -\log \theta_{x|\mathbf{u}} = -\log Pr(\mathbf{x}),$$

and minimizing the penalty – equivalently maximizing the weight, by Definition 11.4 – yields an MPE instantiation.

Problem 11.7. Consider the class of Bayesian networks in which every variable is binary and every nonroot CPT is deterministic (that is, contains only zero/one parameters). Describe a corresponding reduction to WMC with a CNF encoding that includes a single Boolean variable for each network node, no variables for network parameters, and no more than one clause for each network parameter.

Solution. Use one Boolean variable per node, also written X, reading X true as the value x and X false as $\bar{x}$; binarity then already encodes "exactly one value of X is chosen", so no indicator clauses (11.3) and no parameter variables are needed. Write $\ell_x = X$ and $\ell_{\bar{x}} = \neg X$.

Clauses. Determinism makes exactly one of $\theta_{x|\mathbf{u}}, \theta_{\bar{x}|\mathbf{u}}$ zero for each nonroot node X and each parent instantiation $\mathbf{u} = u_1, \ldots, u_m$; for that value v generate the one clause

$$\neg\ell_{u_1} \vee \ldots \vee \neg\ell_{u_m} \vee \neg\ell_v,$$

that is, $\ell_{u_1} \wedge \ldots \wedge \ell_{u_m} \Longrightarrow \neg\ell_v$: the context $\mathbf{u}$ rules out the value v. Roots generate no clause.

Weights. $Wt(X) = \theta_x$ and $Wt(\neg X) = \theta_{\bar{x}}$ at a root, and $Wt(X) = Wt(\neg X) = 1$ at a nonroot.

Size. Exactly n Boolean variables, and exactly one clause of $m + 1$ literals per pair of a nonroot node and a parent instantiation – one per two nonroot parameters, hence strictly fewer than one per network parameter, as required.

Correctness. Truth assignments $\omega_{\mathbf{x}}$ correspond one to one with network instantiations $\mathbf{x}$, and $\omega_{\mathbf{x}}$ violates a clause iff some nonroot factor of $Pr(\mathbf{x}) = \prod_X \theta_{x|\mathbf{u}}$ is zero: the clause for $v\mathbf{u}$ is violated exactly when $v\mathbf{u} \sim \mathbf{x}$ with $\theta_{v|\mathbf{u}} = 0$, and every clause of Δ_N arises this way. (This is not the same as "$\omega_{\mathbf{x}} \models \Delta_N$ iff $Pr(\mathbf{x}) > 0$", since a root parameter may itself be 0.) Every nonroot parameter being 0 or 1, a model therefore has every nonroot factor equal to 1, so

$$Wt(\omega_{\mathbf{x}}) = \prod_{X \text{ root}} \theta_x = Pr(\mathbf{x})$$

whether or not some root parameter vanishes, while a nonmodel has $Pr(\mathbf{x}) = 0$. Nonmodels may thus be added back for free:

$$\text{WMC}(\Delta_N) = \sum_{\omega_{\mathbf{x}} \models \Delta_N} Pr(\mathbf{x}) = \sum_{\mathbf{x}} Pr(\mathbf{x}) = 1,$$

and conjoining the unit clauses $\ell_{e_1}, \ldots, \ell_{e_k}$ for evidence $e = e_1, \ldots, e_k$ retains exactly the models with $\mathbf{x} \sim e$, so

$$\mathrm{WMC}(\Delta_N \wedge \ell_{e_1} \wedge \ldots \wedge \ell_{e_k}) = \sum_{\mathbf{x} \sim e} Pr(\mathbf{x}) = Pr(e),$$

the analogue of Theorem 11.7 here. (Evidence may equally be folded into the weights as in Exercise 11.5.)

For instance, the AND gate $\theta_{x|u_1u_2} = 1$, $\theta_{x|\mathbf{u}} = 0$ otherwise, produces four clauses, one per parent instantiation:

parent instantiation	θ that is 0	clause
$u_1 u_2$	$\theta_{\bar{x}\|u_1u_2}$	$\neg U_1 \vee \neg U_2 \vee X$
$u_1 \bar{u}_2$	$\theta_{x\|u_1\bar{u}_2}$	$\neg U_1 \vee U_2 \vee \neg X$
$\bar{u}_1 u_2$	$\theta_{x\|\bar{u}_1u_2}$	$U_1 \vee \neg U_2 \vee \neg X$
$\bar{u}_1 \bar{u}_2$	$\theta_{x\|\bar{u}_1\bar{u}_2}$	$U_1 \vee U_2 \vee \neg X$

which together are $X \iff U_1 \wedge U_2$, with X carrying weight 1 on both literals.

10.2 Exercises 11.8–11.10

Problem 11.8. Prove Theorem 11.8.

For reference, Theorem 11.8 concerns the *second* CNF encoding of a Bayesian network, given in Section 11.6.2. Let N be a Bayesian network over variables $X_1, \ldots, X_n$. For each network variable X with parents $\mathbf{U}$, fix a total order $x_1 < x_2 < \cdots < x_k$ on the values of X. The encoding uses two kinds of Boolean variables:

- an *indicator* variable I_x for each value x of each network variable X;
- a *parameter* variable $Q_{x|\mathbf{u}}$ for each instantiation $x\mathbf{u}$ of a family $X\mathbf{U}$ in which x is **not** the last value of X in the value order.

The CNF Δ_N consists of the indicator clauses, generated for each variable X with values $x_1, \ldots, x_k$,

$$I_{x_1} \vee I_{x_2} \vee \cdots \vee I_{x_k}, \qquad \neg I_{x_i} \vee \neg I_{x_j} \ \text{ for } i < j, \tag{11.5}$$

together with the parameter clauses, generated for each variable X and each instantiation $\mathbf{u} = u_1, \ldots, u_m$ of its parents $\mathbf{U}$,

$$\begin{aligned} I_{u_1} \wedge \cdots \wedge I_{u_m} \wedge \neg Q_{x_1|\mathbf{u}} \wedge \cdots \wedge \neg Q_{x_{i-1}|\mathbf{u}} \wedge Q_{x_i|\mathbf{u}} \Rightarrow I_{x_i}, \quad i < k, \\ I_{u_1} \wedge \cdots \wedge I_{u_m} \wedge \neg Q_{x_1|\mathbf{u}} \wedge \cdots \wedge \neg Q_{x_{k-1}|\mathbf{u}} \Rightarrow I_{x_k}. \end{aligned} \tag{11.6}$$

Theorem 11.8 states: let N be a Bayesian network inducing probability distribution Pr and let Δ_N be its CNF encoding given by (11.5) and (11.6). For any evidence $e = e_1, \ldots, e_k$ we have

$$Pr(e) \;=\; \mathrm{WMC}(\Delta_N \wedge I_{e_1} \wedge \cdots \wedge I_{e_k}),$$

given the following weights:

$$\begin{aligned} Wt(I_x) &= 1, & Wt(\neg I_x) &= 1, \\ Wt(Q_{x|\mathbf{u}}) &= \frac{\theta_{x|\mathbf{u}}}{1 - \sum_{x'<x} \theta_{x'|\mathbf{u}}}, & Wt(\neg Q_{x|\mathbf{u}}) &= 1 - Wt(Q_{x|\mathbf{u}}). \end{aligned}$$

Solution. The models of Δ_N fall into classes of total weight $Pr(\mathbf{x})$, one class per network instantiation; summing over $\mathbf{x} \sim e$ is then the theorem.

The indicator clauses (11.5) make exactly one I_x true per network variable, so every model ω of Δ_N determines an instantiation $\mathbf{x}(\omega)$, and the classes

$$M(\mathbf{x}) \;=\; \{\, \omega \models \Delta_N \;:\; \mathbf{x}(\omega) = \mathbf{x} \,\}$$

are disjoint and exhaust the models.

Fix $\mathbf{x}$. In $M(\mathbf{x})$ the indicators are pinned and weigh 1, so a model's weight is the product over its pa-

rameter literals; these split into blocks $Q_{x_1|\mathbf{u}}, \ldots, Q_{x_{k-1}|\mathbf{u}}$, one per family $X\mathbf{U}$ and parent instantiation $\mathbf{u}$, each block mentioned only by its own clauses of (11.6). Abbreviate $\theta_i = \theta_{x_i|\mathbf{u}}$, $S_i = \sum_{j<i} \theta_j$, and $\lambda_i = Wt(Q_{x_i|\mathbf{u}}) = \theta_i/(1 - S_i)$; a CPT row sums to 1, so $S_i + \theta_i \leq 1$ and $0 \leq \lambda_i \leq 1$, legitimate weights, with the convention $\lambda_i = 0$ when $S_i = 1$.

(i) $\mathbf{u} \not\sim \mathbf{x}$. Some I_{u_j} is false, so every antecedent in (11.6) for this $\mathbf{u}$ fails, the block is free, and its contribution factorizes to $\prod_{i<k}\big(\lambda_i + (1 - \lambda_i)\big) = 1$.

(ii) $\mathbf{u} \sim \mathbf{x}$, with $\mathbf{x}$ setting $X = x_r$. Let f be the block's first true index, $f = \min\{i < k : Q_{x_i|\mathbf{u}} \text{ true}\}$ and $f = k$ if there is none. The clause of index $i < k$ has a true antecedent precisely when $f = i$, and the last clause precisely when $f = k$, while I_{x_i} is true only for $i = r$; hence the group is satisfied iff $f = r$. Admissibility thus fixes $\neg Q_{x_1|\mathbf{u}}, \ldots, \neg Q_{x_{r-1}|\mathbf{u}}$ and, when $r < k$, also $Q_{x_r|\mathbf{u}}$, leaving a free tail that sums to 1 as in (i). Since $1 - \lambda_j = (1 - S_{j+1})/(1 - S_j)$ telescopes to $\prod_{j<i}(1 - \lambda_j) = 1 - S_i$, the block contributes

$$\lambda_r \prod_{j<r}(1 - \lambda_j) = \frac{\theta_r}{1 - S_r}(1 - S_r) = \theta_r \quad (r < k), \qquad \prod_{j<k}(1 - \lambda_j) = 1 - S_k = \theta_k \quad (r = k),$$

that is, $\theta_{x|\mathbf{u}}$ with $x\mathbf{u} \sim \mathbf{x}$ in both cases. (If $S_i = 1$ for a least i, then $\prod_{j<i}(1 - \lambda_j) = 1 - S_i = 0$ and $\theta_r = 0$ for all $r \geq i$, so both sides of each identity vanish.)

Each family contributes exactly one block of type (ii), namely that of its unique parent instantiation compatible with $\mathbf{x}$, so multiplying over all blocks and applying the chain rule (Chapter 3),

$$\sum_{\omega \in M(\mathbf{x})} Wt(\omega) \;=\; \prod_{\theta_{x|\mathbf{u}} \sim \mathbf{x}} \theta_{x|\mathbf{u}} \;=\; Pr(\mathbf{x}).$$

Finally, a model of $\Delta = \Delta_N \wedge I_{e_1} \wedge \cdots \wedge I_{e_k}$ is a model of Δ_N with $\mathbf{x}(\omega) \sim e$, one indicator per variable being true, so the models of Δ are the disjoint union of the $M(\mathbf{x})$ over $\mathbf{x} \sim e$ and

$$\mathrm{WMC}(\Delta) = \sum_{\mathbf{x} \sim e} \sum_{\omega \in M(\mathbf{x})} Wt(\omega) = \sum_{\mathbf{x} \sim e} Pr(\mathbf{x}) = Pr(e). \qquad \blacksquare$$

Problem 11.9. Show how we can extend Theorems 11.7 and 11.8 to handle a more general type of evidence in which we may not know the exact value of a multivalued variable but only that some of its values are impossible.

For reference: Theorem 11.7 concerns the first CNF encoding of Section 11.6.1, which uses an indicator variable I_x for each value x of each network variable X and a parameter variable $P_{x|\mathbf{u}}$ for each family instantiation $x\mathbf{u}$, with indicator clauses

$$I_{x_1} \vee \cdots \vee I_{x_k}, \qquad \neg I_{x_i} \vee \neg I_{x_j} \;\text{ for } i < j, \tag{11.3}$$

and parameter clauses

$$I_{u_1} \wedge \cdots \wedge I_{u_m} \wedge I_x \iff P_{x|u_1,\ldots,u_m}. \tag{11.4}$$

It states that $Pr(e) = \mathrm{WMC}(\Delta_N \wedge I_{e_1} \wedge \cdots \wedge I_{e_k})$ under the weights $Wt(I_x) = Wt(\neg I_x) = Wt(\neg P_{x|\mathbf{u}}) = 1$ and $Wt(P_{x|\mathbf{u}}) = \theta_{x|\mathbf{u}}$. Theorem 11.8 states the analogous identity for the second encoding of Section 11.6.2, given by the indicator clauses (11.5) — which are identical to (11.3) — and the parameter clauses (11.6), under the weights $Wt(I_x) = Wt(\neg I_x) = 1$, $Wt(Q_{x|\mathbf{u}}) = \theta_{x|\mathbf{u}}/(1 - \sum_{x'<x} \theta_{x'|\mathbf{u}})$ and $Wt(\neg Q_{x|\mathbf{u}}) = 1 - Wt(Q_{x|\mathbf{u}})$.

In both theorems the evidence $e = e_1, \ldots, e_k$ is an instantiation: it pins down the exact value of each of k variables. The task is to handle instead evidence of the form "variable X is known not to take any of the values $x \notin \Lambda_X$," for an arbitrary set Λ_X of surviving values.

Solution. Replace the evidence unit clauses by the disjunction of the surviving indicators: writing $\Lambda_{E_i} \subseteq$

val(E_i) for the values of the observed variable E_i that have not been ruled out and $e = \bigwedge_{i=1}^{k}(E_i \in \Lambda_{E_i})$,

$$\Delta_N^e \;=\; \Delta_N \wedge \bigwedge_{i=1}^{k}\Big(\bigvee_{x\in\Lambda_{E_i}} I_x\Big), \qquad Pr(e) \;=\; \mathrm{WMC}(\Delta_N^e),$$

with the weights of Theorem 11.7 or Theorem 11.8 left unchanged.
Both theorems rest on two facts established in their proofs (Exercise 11.8 for the second encoding): every model ω of Δ_N makes exactly one indicator true per network variable, by the clauses (11.3) = (11.5), and so determines an instantiation $\mathbf{x}(\omega)$; and the classes $M(\mathbf{x}) = \{\omega \models \Delta_N : \mathbf{x}(\omega) = \mathbf{x}\}$ partition the models with $\sum_{\omega\in M(\mathbf{x})} Wt(\omega) = Pr(\mathbf{x})$ – the latter by applying the theorem with the complete instantiation $\mathbf{x}$ as the evidence. Since ω makes exactly one indicator of E_i true, it satisfies the added clause iff $\mathbf{x}(\omega)$ gives E_i a value in Λ_{E_i}; hence $\omega \models \Delta_N^e$ iff $\omega \models \Delta_N$ and $\mathbf{x}(\omega) \models e$, and

$$\mathrm{WMC}(\Delta_N^e) = \sum_{\mathbf{x}\models e}\;\sum_{\omega\in M(\mathbf{x})} Wt(\omega) = \sum_{\mathbf{x}\models e} Pr(\mathbf{x}) = Pr(e),$$

the last step being the definition of the probability of an event. Only the indicator clauses were used, and the two encodings share them. Singleton Λ_{E_i} recovers the two theorems verbatim; $\Lambda_{E_i} = \emptyset$ makes the CNF unsatisfiable, correctly returning $0 = Pr(e)$; and modulo Δ_N the added clause is equivalent to the unit clauses $\neg I_x$ for $x \notin \Lambda_{E_i}$, which a model counter propagates away.
Method (2): leave Δ_N untouched and change only the indicator weights, as in Exercise 11.5 – $Wt_e(I_x) = 0$ when x is a ruled-out value of an observed variable and 1 otherwise, $Wt_e(\neg I_x) = 1$, parameter weights unchanged. A model $\omega \in M(\mathbf{x})$ carries exactly one positive indicator per variable, so $Wt_e(\omega) = Wt(\omega)$ when $\mathbf{x} \models e$ and 0 otherwise, and summing by class gives $Pr(e)$ again. This is the version to use when one compiled structure must answer many evidence sets, only the leaf weights changing.

Problem 11.10. Show that we can drop the indicator clauses given in (11.9) without affecting the correctness of the MPE to W-MAXSAT reduction given in Section 11.7.
For reference: in the reduction of Section 11.7 a Bayesian network N over variables $X_1, \ldots, X_n$ is encoded as a weighted CNF whose only Boolean variables are the indicators I_x, one for each value x of each network variable X. For each network variable X with values $x_1, \ldots, x_k$ we generate the hard indicator clauses

$$(I_{x_1} \vee I_{x_2} \vee \cdots \vee I_{x_k})^W \tag{11.8}$$

$$(\neg I_{x_i} \vee \neg I_{x_j})^W, \qquad \text{for } i < j, \tag{11.9}$$

and for each network parameter $\theta_{x|u_1,\ldots,u_m}$ we generate the parameter clause

$$(\neg I_x \vee \neg I_{u_1} \vee \cdots \vee \neg I_{u_m})^{-\log \theta_{x|u_1,\ldots,u_m}}, \tag{11.10}$$

where $-\log 0$ is defined to be W, so a zero parameter also yields a hard clause. The special weight W is chosen greater than the sum of the weights of all soft clauses. Definition 11.4 poses W-MAXSAT as the problem of finding a truth assignment of maximal weight $Wt(\hat{\omega}) = \sum_{\hat{\omega}\models\alpha_i} w_i$. Section 11.7 then defines the penalty of a truth assignment as the total weight of the clauses it violates, $Pn(\hat{\omega}) = \sum_{\hat{\omega}\not\models\alpha_i} w_i$, and notes that $Wt(\hat{\omega}) + Pn(\hat{\omega})$ is the constant sum of all clause weights, so W-MAXSAT is equivalently the problem of finding an assignment of minimum penalty. A truth assignment $\hat{\omega}$ *corresponds* to a network instantiation $x_1, \ldots, x_n$ when $I_{x_1}, \ldots, I_{x_n}$ appear positively in $\hat{\omega}$ and all other indicators appear negatively; Section 11.7 shows that such an assignment has penalty $-\log Pr(x_1, \ldots, x_n)$.
Show that if the clauses (11.9) are deleted from the encoding — keeping only (11.8) and (11.10) — then a minimum-penalty truth assignment still yields an MPE instantiation.

Solution. Deleting (11.9) is harmless because the surviving CNF Φ – the hard clauses (11.8) together with the parameter clauses (11.10) – has no clause of mixed polarity: (11.8) is all positive indicators

and (11.10) all negated ones. So turning an indicator from true to false can only satisfy more parameter clauses, and extra true indicators never pay. (The sum of soft weights is unchanged by deleting hard clauses, so the same W serves.)
(i) The minimum penalty is below W. Since $\sum_{\mathbf{x}} Pr(\mathbf{x}) = 1$, some $\mathbf{x}^0$ has $Pr(\mathbf{x}^0) > 0$; its corresponding assignment $\hat{\omega}^0$ satisfies every clause (11.8), and every hard clause (11.10) as well – such a clause has $\theta_{x|\mathbf{u}} = 0$ and is violated only when that parameter is compatible with $\mathbf{x}^0$, which would force $Pr(\mathbf{x}^0) = 0$. Hence $Pn(\hat{\omega}^0) = -\log Pr(\mathbf{x}^0) < W$, so any minimum-penalty $\hat{\omega}$ satisfies every hard clause and in particular makes at least one indicator true per network variable.
(ii) Pruning extra true indicators never costs. Given such an $\hat{\omega}$, choose one true indicator I_{x^*} per variable and let $\hat{\omega}'$ set every other indicator false. Every clause satisfied by $\hat{\omega}$ is still satisfied: a clause (11.8) by I_{x^*}, which stays true; a clause (11.10) by whichever literal $\neg I_y$ satisfied it, since indicators only move from true to false. So $Pn(\hat{\omega}') \leq Pn(\hat{\omega}) = p^*$, whence $\hat{\omega}'$ is optimal too, and it corresponds to an instantiation $\mathbf{x}^*$.
(iii) The penalty of a corresponding assignment, as computed in Section 11.7 without ever using (11.9): a clause (11.10) is violated by $\hat{\omega}'$ exactly when its parameter is compatible with $\mathbf{x}$, so by the chain rule

$$Pn(\hat{\omega}') = \sum_{\theta_{x|\mathbf{u}} \sim \mathbf{x}} -\log \theta_{x|\mathbf{u}} = -\log Pr(\mathbf{x}).$$

Under the convention $-\log 0 = W$ this is an equality of symbols when $Pr(\mathbf{x}) = 0$ (with t zero compatible parameters the penalty is tW), but all that is used below is

$$\begin{aligned} Pr(\mathbf{x}) > 0 \;&\Rightarrow\; Pn(\hat{\omega}') = -\log Pr(\mathbf{x}) < W, \\ Pr(\mathbf{x}) = 0 \;&\Rightarrow\; Pn(\hat{\omega}') \geq W. \end{aligned}$$

Since $p^* < W$ by (i), (iii) gives $Pr(\mathbf{x}^*) > 0$ and $p^* = -\log Pr(\mathbf{x}^*)$. Every instantiation $\mathbf{x}$ has a corresponding assignment competing for the minimum, so $-\log Pr(\mathbf{x}) \geq p^*$ whenever $Pr(\mathbf{x}) > 0$, that is $Pr(\mathbf{x}) \leq Pr(\mathbf{x}^*)$, and trivially so when $Pr(\mathbf{x}) = 0$. Hence $\mathbf{x}^*$ is an MPE instantiation and $p^* = -\log \max_{\mathbf{x}} Pr(\mathbf{x})$, the guarantee of the original encoding. The choice function in (ii) was arbitrary, so any selection of one true indicator per variable decodes a minimum-penalty assignment to an MPE. Evidence asserted by hard unit clauses $I_{e_1}, \ldots, I_{e_k}$ changes nothing, provided $Pr(e) > 0$ so that (i) still finds a positive-probability $\mathbf{x}^0 \sim e$: take I_{e_i} as the chosen indicator of E_i in (ii), and $\hat{\omega}'$ still satisfies the unit clauses.

11 Compiling Bayesian Networks

Contents

11.1 Exercises 12.1–12.7 175
11.2 Exercises 12.8–12.14 182
11.3 Exercises 12.15–12.20 189

11.1 Exercises 12.1–12.7

Problem 12.1. Consider the arithmetic circuit in Figure 12.13. The circuit is built over three binary variables A, B, C (values a_1, a_2, b_1, b_2, c_1, c_2) and it is the compiled form of the naive-Bayes network $B \leftarrow A \rightarrow C$. Its leaves are the indicators $\lambda_{a_1}, \lambda_{a_2}, \lambda_{b_1}, \lambda_{b_2}, \lambda_{c_1}, \lambda_{c_2}$ and the parameters, whose values are printed in the figure:

parameter	value	parameter	value
θ_{a_1}	.1	θ_{a_2}	.9
$\theta_{b_1\|a_1}$	.2	$\theta_{b_2\|a_1}$	.8
$\theta_{b_1\|a_2}$	.3	$\theta_{b_2\|a_2}$	.7
$\theta_{c_1\|a_1}$	.4	$\theta_{c_2\|a_1}$	.6
$\theta_{c_1\|a_2}$	.01	$\theta_{c_2\|a_2}$	.99

Reading the figure bottom-up, the internal nodes of the circuit are (each node is written as its operation applied to its children):

node	definition
n_1	$\star(\lambda_{c_1}, \theta_{c_1\|a_1})$
n_2	$\star(\lambda_{c_2}, \theta_{c_2\|a_1})$
n_3	$\star(\lambda_{a_1}, \theta_{a_1})$
n_4	$\star(\lambda_{a_2}, \theta_{a_2})$
n_5	$\star(\lambda_{c_1}, \theta_{c_1\|a_2})$
n_6	$\star(\lambda_{c_2}, \theta_{c_2\|a_2})$
n_7	$\star(\theta_{b_1\|a_1}, n_3)$
n_8	$\star(\theta_{b_1\|a_2}, n_4)$
n_9	$+(n_1, n_2)$
n_{10}	$+(n_5, n_6)$
n_{11}	$\star(\theta_{b_2\|a_1}, n_3)$
n_{12}	$\star(\theta_{b_2\|a_2}, n_4)$
n_{13}	$\star(n_7, n_9)$
n_{14}	$\star(n_8, n_{10})$
n_{15}	$\star(n_9, n_{11})$
n_{16}	$\star(n_{10}, n_{12})$
n_{17}	$+(n_{13}, n_{14})$
n_{18}	$+(n_{15}, n_{16})$
n_{19}	$\star(\lambda_{b_1}, n_{17})$
n_{20}	$\star(\lambda_{b_2}, n_{18})$
n_{21}	$+(n_{19}, n_{20})$

with n_{21} the circuit output (root). Here $\star$ denotes a multiplication node and $+$ an addition node, each listed with its children.

(a) Construct the polynomial f represented by this arithmetic circuit.

(b) Compute the partial derivatives $\partial f/\partial\lambda_{a_1}$, $\partial f/\partial\lambda_{a_2}$, $\partial f/\partial\theta_{c_1|a_1}$, and $\partial f/\partial\theta_{c_2|a_2}$.

(c) Evaluate the derivatives in (b) at evidence $e = a_1 c_1$.

(d) What is the probabilistic meaning of these derivatives?

Solution. (a) Expanding from the root, $n_{21} = \lambda_{b_1} n_{17} + \lambda_{b_2} n_{18}$ with $n_{13}, \ldots, n_{16}$ pairing $\theta_{b|a} n_3$ or $\theta_{b|a} n_4$ with the matching C-sum $n_9 = \sum_c \lambda_c \theta_{c|a_1}$, $n_{10} = \sum_c \lambda_c \theta_{c|a_2}$, so

$$f \;=\; \sum_b \lambda_b \sum_a \lambda_a \theta_a \theta_{b|a} \Big(\sum_c \lambda_c \theta_{c|a} \Big).$$

Multiplied out this is the network polynomial of $B \leftarrow A \rightarrow C$ in the canonical form of Definition 12.1, one term per instantiation of ABC:

$$
\begin{aligned}
f = {} & \lambda_{a_1}\lambda_{b_1}\lambda_{c_1}\,\theta_{a_1}\theta_{b_1|a_1}\theta_{c_1|a_1} \\
& + \lambda_{a_1}\lambda_{b_1}\lambda_{c_2}\,\theta_{a_1}\theta_{b_1|a_1}\theta_{c_2|a_1} \\
& + \lambda_{a_1}\lambda_{b_2}\lambda_{c_1}\,\theta_{a_1}\theta_{b_2|a_1}\theta_{c_1|a_1} \\
& + \lambda_{a_1}\lambda_{b_2}\lambda_{c_2}\,\theta_{a_1}\theta_{b_2|a_1}\theta_{c_2|a_1} \\
& + \lambda_{a_2}\lambda_{b_1}\lambda_{c_1}\,\theta_{a_2}\theta_{b_1|a_2}\theta_{c_1|a_2} \\
& + \lambda_{a_2}\lambda_{b_1}\lambda_{c_2}\,\theta_{a_2}\theta_{b_1|a_2}\theta_{c_2|a_2} \\
& + \lambda_{a_2}\lambda_{b_2}\lambda_{c_1}\,\theta_{a_2}\theta_{b_2|a_2}\theta_{c_1|a_2} \\
& + \lambda_{a_2}\lambda_{b_2}\lambda_{c_2}\,\theta_{a_2}\theta_{b_2|a_2}\theta_{c_2|a_2}.
\end{aligned}
$$

(b) The polynomial is multilinear, so each derivative deletes the differentiation variable from the terms containing it and discards the rest:

$$
\begin{aligned}
\frac{\partial f}{\partial \lambda_{a_i}} &= \theta_{a_i}\big(\lambda_{b_1}\theta_{b_1|a_i} + \lambda_{b_2}\theta_{b_2|a_i}\big)\big(\lambda_{c_1}\theta_{c_1|a_i} + \lambda_{c_2}\theta_{c_2|a_i}\big), \quad i = 1, 2, \\
\frac{\partial f}{\partial \theta_{c_1|a_1}} &= \lambda_{a_1}\lambda_{c_1}\theta_{a_1}\big(\lambda_{b_1}\theta_{b_1|a_1} + \lambda_{b_2}\theta_{b_2|a_1}\big), \\
\frac{\partial f}{\partial \theta_{c_2|a_2}} &= \lambda_{a_2}\lambda_{c_2}\theta_{a_2}\big(\lambda_{b_1}\theta_{b_1|a_2} + \lambda_{b_2}\theta_{b_2|a_2}\big).
\end{aligned}
$$

(c) Evidence $e = a_1c_1$ sets $\lambda_{a_1} = \lambda_{c_1} = 1$, $\lambda_{a_2} = \lambda_{c_2} = 0$, and $\lambda_{b_1} = \lambda_{b_2} = 1$ since B is unmentioned, so

$$
\begin{aligned}
\frac{\partial f}{\partial \lambda_{a_1}}(e) &= (.1)(.2 + .8)(.4) = .04, \qquad \frac{\partial f}{\partial \lambda_{a_2}}(e) = (.9)(.3 + .7)(.01) = .009, \\
\frac{\partial f}{\partial \theta_{c_1|a_1}}(e) &= (1)(1)(.1)(.2 + .8) = .1, \quad \frac{\partial f}{\partial \theta_{c_2|a_2}}(e) = (0)(0)(.9)(1) = 0.
\end{aligned}
$$

(d) By Theorem 12.2, $\partial f/\partial \lambda_x(e) = \Pr(x, e - X)$ and $\theta_{x|u}\,\partial f/\partial \theta_{x|u}(e) = \Pr(x, u, e)$, and here $e - A = c_1$, $e - C = a_1$:

- $.04 = \Pr(a_1c_1) = \Pr(e)$, since a_1 is already the value of A in e.
- $.009 = \Pr(a_2c_1)$, the probability of the evidence with A flipped to a_2, read off without re-evaluating the circuit.
- $(.4)(.1) = .04 = \Pr(c_1, a_1, e)$, the family marginal of CA; equivalently, by (12.6), .1 is the rate $\partial \Pr(e)/\partial \theta_{c_1|a_1}$ used in Chapter 16.
- $0 = \Pr(c_2, a_2, e)$, necessarily zero because c_2a_2 contradicts e.

Problem 12.2. Consider the arithmetic circuit in Figure 12.13 (its nodes, structure and parameter values are transcribed in Exercise 12.1). Evaluate and differentiate this circuit at evidence $e = a_1c_1$. Perform the differentiation using both propagation schemes given by Algorithms 34 and 35.

Solution. Evidence $e = a_1c_1$ sets $\lambda_{a_1} = \lambda_{c_1} = 1$, $\lambda_{a_2} = \lambda_{c_2} = 0$, $\lambda_{b_1} = \lambda_{b_2} = 1$. Visiting children before parents gives the bottom-up pass shared by both algorithms:

node	expression	value
n_1	$\lambda_{c_1}\theta_{c_1\|a_1} = (1)(.4)$	.4
n_2	$\lambda_{c_2}\theta_{c_2\|a_1} = (0)(.6)$	0
n_3	$\lambda_{a_1}\theta_{a_1} = (1)(.1)$	.1
n_4	$\lambda_{a_2}\theta_{a_2} = (0)(.9)$	0
n_5	$\lambda_{c_1}\theta_{c_1\|a_2} = (1)(.01)$	.01
n_6	$\lambda_{c_2}\theta_{c_2\|a_2} = (0)(.99)$	0
n_7	$\theta_{b_1\|a_1}n_3 = (.2)(.1)$	.02
n_8	$\theta_{b_1\|a_2}n_4 = (.3)(0)$	0
n_9	$n_1 + n_2 = .4 + 0$	.4
n_{10}	$n_5 + n_6 = .01 + 0$	.01
n_{11}	$\theta_{b_2\|a_1}n_3 = (.8)(.1)$	.08
n_{12}	$\theta_{b_2\|a_2}n_4 = (.7)(0)$	0
n_{13}	$n_7n_9 = (.02)(.4)$	.008
n_{14}	$n_8n_{10} = (0)(.01)$	0
n_{15}	$n_9n_{11} = (.4)(.08)$	.032
n_{16}	$n_{10}n_{12} = (.01)(0)$	0
n_{17}	$n_{13} + n_{14}$	.008
n_{18}	$n_{15} + n_{16}$	.032
n_{19}	$\lambda_{b_1}n_{17} = (1)(.008)$	.008
n_{20}	$\lambda_{b_2}n_{18} = (1)(.032)$	.032
n_{21}	$n_{19} + n_{20} = .008 + .032$	.04

The output is $\mathrm{vr}(n_{21}) = .04 = \Pr(a_1c_1)$, as Theorem 12.1 requires.

Algorithm 34. Set $\mathrm{dr}(n_{21}) = 1$, all other registers 0, and visit parents before children, a node v adding $\mathrm{dr}(p)$ from an addition parent p and $\mathrm{dr}(p)\prod_{v'\neq v}\mathrm{vr}(v')$ from a multiplication parent. Note n_9 has parents n_{13}, n_{15}, and n_3 has parents n_7, n_{11} (symmetrically for n_{10}, n_4):

$$\begin{aligned}
&\mathrm{dr}(n_{19}) = \mathrm{dr}(n_{20}) = \mathrm{dr}(n_{17}) = \mathrm{dr}(n_{18}) = 1,\\
&\mathrm{dr}(n_{13}) = \mathrm{dr}(n_{14}) = \mathrm{dr}(n_{15}) = \mathrm{dr}(n_{16}) = 1,\\
&\mathrm{dr}(\lambda_{b_1}) = (1)(.008) = .008, \quad \mathrm{dr}(\lambda_{b_2}) = (1)(.032) = .032,\\
&\mathrm{dr}(n_7) = \mathrm{dr}(n_{11}) = (1)(.4) = .4, \quad \mathrm{dr}(n_8) = \mathrm{dr}(n_{12}) = (1)(.01) = .01,\\
&\mathrm{dr}(n_9) = (1)(.02) + (1)(.08) = .1, \quad \mathrm{dr}(n_{10}) = (1)(0) + (1)(0) = 0,\\
&\mathrm{dr}(n_1) = \mathrm{dr}(n_2) = .1, \quad \mathrm{dr}(n_5) = \mathrm{dr}(n_6) = 0,\\
&\mathrm{dr}(n_3) = (.4)(.2) + (.4)(.8) = .4, \quad \mathrm{dr}(n_4) = (.01)(.3) + (.01)(.7) = .01.
\end{aligned}$$

The leaves follow, λ_{c_1} having parents n_1, n_5 and λ_{c_2} parents n_2, n_6:

leaf	computation	dr
λ_{a_1}	$\mathrm{dr}(n_3)\theta_{a_1} = (.4)(.1)$	.04
θ_{a_1}	$\mathrm{dr}(n_3)\lambda_{a_1} = (.4)(1)$	.4
λ_{a_2}	$\mathrm{dr}(n_4)\theta_{a_2} = (.01)(.9)$	.009
θ_{a_2}	$\mathrm{dr}(n_4)\lambda_{a_2} = (.01)(0)$	0
λ_{b_1}	(above)	.008
λ_{b_2}	(above)	.032
$\theta_{b_1\|a_1}$	$\mathrm{dr}(n_7)\mathrm{vr}(n_3) = (.4)(.1)$	.04
$\theta_{b_2\|a_1}$	$\mathrm{dr}(n_{11})\mathrm{vr}(n_3) = (.4)(.1)$	.04
$\theta_{b_1\|a_2}$	$\mathrm{dr}(n_8)\mathrm{vr}(n_4) = (.01)(0)$	0
$\theta_{b_2\|a_2}$	$\mathrm{dr}(n_{12})\mathrm{vr}(n_4) = (.01)(0)$	0
λ_{c_1}	from n_1 and n_5: $(.1)(.4) + (0)(.01)$	.04
λ_{c_2}	from n_2 and n_6: $(.1)(.6) + (0)(.99)$	.06
$\theta_{c_1\|a_1}$	$\mathrm{dr}(n_1)\lambda_{c_1} = (.1)(1)$	.1
$\theta_{c_2\|a_1}$	$\mathrm{dr}(n_2)\lambda_{c_2} = (.1)(0)$	0
$\theta_{c_1\|a_2}$	$\mathrm{dr}(n_5)\lambda_{c_1} = (0)(1)$	0
$\theta_{c_2\|a_2}$	$\mathrm{dr}(n_6)\lambda_{c_2} = (0)(0)$	0

Algorithm 35 replaces $\prod_{v' \neq v} \mathrm{vr}(v')$ by the division $\mathrm{vr}(p)/\mathrm{vr}(v)$, carrying one bit per multiplication node: $\mathrm{bit}(p) = 1$ exactly when one child of p is zero, and then $\mathrm{vr}(p)$ stores the product of the remaining children. The algorithm is stated for strictly alternating circuits, which this one is not (n_7, n_8, n_{11}, n_{12} and $n_{13}, \ldots, n_{16}$ have multiplication children), so use the relaxation the book allows, testing a child's type as well as its bit: the true value of c is 0 when c is a multiplication node with $\mathrm{bit}(c) = 1$, and $\mathrm{vr}(c)$ otherwise. An addition node then sums only the children of bit 0.

node	bit	vr	true value
n_1	0	.4	.4
n_2	1	.6	0
n_3	0	.1	.1
n_4	1	.9	0
n_5	0	.01	.01
n_6	1	.99	0
n_7	0	.02	.02
n_8	1	.3	0
n_9	(add)	.4	.4
n_{10}	(add)	.01	.01
n_{11}	0	.08	.08
n_{12}	1	.7	0
n_{13}	0	.008	.008
n_{14}	1	.01	0
n_{15}	0	.032	.032
n_{16}	1	.01	0
n_{17}	(add)	.008	.008
n_{18}	(add)	.032	.032
n_{19}	0	.008	.008
n_{20}	0	.032	.032
n_{21}	(add)	.04	.04

The top-down pass adds, at a multiplication parent p of v: nothing if $\mathrm{vr}(p) = 0$; $\mathrm{dr}(p)\mathrm{vr}(p)/\mathrm{vr}(v)$ if $\mathrm{bit}(p) = 0$; and $\mathrm{dr}(p)\mathrm{vr}(p)$ if $\mathrm{bit}(p) = 1$ and v is the single zero-valued child of p, nothing otherwise. From $\mathrm{dr}(n_{21}) = 1$:

$$
\begin{aligned}
&\mathrm{dr}(n_{19}) = \mathrm{dr}(n_{20}) = 1,\\
&\mathrm{dr}(\lambda_{b_1}) = (1)(.008)/1 = .008, \quad \mathrm{dr}(n_{17}) = (1)(.008)/.008 = 1,\\
&\mathrm{dr}(\lambda_{b_2}) = (1)(.032)/1 = .032, \quad \mathrm{dr}(n_{18}) = (1)(.032)/.032 = 1,\\
&\mathrm{dr}(n_{13}) = \mathrm{dr}(n_{14}) = \mathrm{dr}(n_{15}) = \mathrm{dr}(n_{16}) = 1,\\
&\mathrm{dr}(n_7) = (1)(.008)/.02 = .4, \quad \mathrm{dr}(n_{11}) = (1)(.032)/.08 = .4,\\
&\mathrm{dr}(n_9) = (1)(.008)/.4 + (1)(.032)/.4 = .1,\\
&\mathrm{dr}(n_8) = (1)(.01) = .01, \quad \mathrm{dr}(n_{12}) = (1)(.01) = .01, \quad \mathrm{dr}(n_{10}) = 0,\\
&\mathrm{dr}(n_1) = \mathrm{dr}(n_2) = .1, \quad \mathrm{dr}(n_5) = \mathrm{dr}(n_6) = 0,\\
&\mathrm{dr}(n_3) = (.4)(.02)/.1 + (.4)(.08)/.1 = .4, \quad \mathrm{dr}(n_4) = (.01)(.3) + (.01)(.7) = .01,
\end{aligned}
$$

where $\mathrm{dr}(n_{10}) = 0$ because the single zero-valued child of n_{14} is n_8 and that of n_{16} is n_{12}, so the bit-1 branch never fires for n_{10}, while n_4 is the zero child of both n_8 and n_{12}, so it does fire there. The leaves:

$$\begin{aligned}
&\mathrm{dr}(\theta_{b_1|a_1}) = (.4)(.02)/.2 = .04, \quad \mathrm{dr}(\theta_{b_2|a_1}) = (.4)(.08)/.8 = .04,\\
&\mathrm{dr}(\theta_{b_1|a_2}) = \mathrm{dr}(\theta_{b_2|a_2}) = 0,\\
&\quad \mathrm{dr}(\lambda_{a_1}) = (.4)(.1)/1 = .04, \quad \mathrm{dr}(\theta_{a_1}) = (.4)(.1)/.1 = .4,\\
&\quad \mathrm{dr}(\lambda_{a_2}) = (.01)(.9) = .009, \quad \mathrm{dr}(\theta_{a_2}) = 0,\\
&\quad \mathrm{dr}(\lambda_{c_1}) = (.1)(.4)/1 + (0)(.01)/1 = .04, \quad \mathrm{dr}(\lambda_{c_2}) = (.1)(.6) = .06,\\
&\mathrm{dr}(\theta_{c_1|a_1}) = (.1)(.4)/.4 = .1, \quad \mathrm{dr}(\theta_{c_2|a_1}) = 0,\\
&\mathrm{dr}(\theta_{c_1|a_2}) = (0)(.01)/.01 = 0, \quad \mathrm{dr}(\theta_{c_2|a_2}) = 0.
\end{aligned}$$

Both parents of λ_{c_2} have bit 1 with λ_{c_2} as their single zero child, contributing $(.1)(.6)$ and $(0)(.99)$, whereas $\theta_{c_2|a_1}$ gets nothing from n_2, not being that child – exactly the case a naive division scheme would mishandle, since $\mathrm{vr}(\lambda_{c_2}) = 0$. Every register matches Algorithm 34, which computes the same $\mathrm{dr}(v) = \partial \mathrm{vr}(r)/\partial \mathrm{vr}(v)$ but forms the sibling products explicitly.

Problem 12.3. Consider a Bayesian network with structure $A \to B \to C$, where all variables are binary (values a_1, a_2; b_1, b_2; c_1, c_2). Construct an arithmetic circuit for this network using the method of variable elimination given in Section 12.4.1 and using elimination order C, B, A.

Solution. Run variable elimination on circuit factors (Section 12.4.1), every factor entry being a circuit node and $\star$, $+$ creating a multiplication resp. addition node with the given children. The initial factor of a family $X\mathbf{U}$ maps $x\mathbf{u}$ to $\star(\lambda_x, \theta_{x|u})$, one per CPT:

A	f_A
a_1	$n_1 = \star(\lambda_{a_1}, \theta_{a_1})$
a_2	$n_2 = \star(\lambda_{a_2}, \theta_{a_2})$

A	B	$f_{B\|A}$
a_1	b_1	$n_3 = \star(\lambda_{b_1}, \theta_{b_1\|a_1})$
a_1	b_2	$n_4 = \star(\lambda_{b_2}, \theta_{b_2\|a_1})$
a_2	b_1	$n_5 = \star(\lambda_{b_1}, \theta_{b_1\|a_2})$
a_2	b_2	$n_6 = \star(\lambda_{b_2}, \theta_{b_2\|a_2})$

B	C	$f_{C\|B}$
b_1	c_1	$n_7 = \star(\lambda_{c_1}, \theta_{c_1\|b_1})$
b_1	c_2	$n_8 = \star(\lambda_{c_2}, \theta_{c_2\|b_1})$
b_2	c_1	$n_9 = \star(\lambda_{c_1}, \theta_{c_1\|b_2})$
b_2	c_2	$n_{10} = \star(\lambda_{c_2}, \theta_{c_2\|b_2})$

Leaves are shared – λ_{b_1} by n_3 and n_5, λ_{c_1} by n_7 and n_9 – which is what keeps the circuit compact. Eliminating C from $f_{C|B}$, the only factor mentioning it, makes one addition node per value of B:

B	$\sum_C f_{C\|B}$
b_1	$n_{11} = +(n_7, n_8)$
b_2	$n_{12} = +(n_9, n_{10})$

Eliminating B: multiply that factor by $f_{B|A}$,

A	B	product
a_1	b_1	$n_{13} = \star(n_3, n_{11})$
a_1	b_2	$n_{14} = \star(n_4, n_{12})$
a_2	b_1	$n_{15} = \star(n_5, n_{11})$
a_2	b_2	$n_{16} = \star(n_6, n_{12})$

and sum B out:

A	result
a_1	$n_{17} = +(n_{13}, n_{14})$
a_2	$n_{18} = +(n_{15}, n_{16})$

Eliminating A: multiply by f_A,

A	product
a_1	$n_{19} = \star(n_1, n_{17})$
a_2	$n_{20} = \star(n_2, n_{18})$

and sum A out, leaving the trivial factor whose single entry

$$n_{21} \;=\; +(n_{19}, n_{20})$$

is the circuit root. So the circuit has 16 leaves (six indicators, ten parameters), 16 multiplication nodes (n_1 to n_{10}, n_{13} to n_{16}, n_{19}, n_{20}) and 5 addition nodes ($n_{11}, n_{12}, n_{17}, n_{18}, n_{21}$), every internal node binary, hence 42 edges. Unwinding the root gives $\sum_{a,b,c} \lambda_a \lambda_b \lambda_c \, \theta_a \theta_{b|a} \theta_{c|b}$ (Check!), the network polynomial of Definition 12.1, so the root evaluates to $\Pr(e)$ for every e by Theorem 12.1. The order C, B, A has width $w = 1$, matching the $O(n \exp(w))$ size bound of Section 12.4.1; the compression relative to the eight-term polynomial comes from the sharing of n_{11}, n_{12} and of the leaves.

Problem 12.4. Show that the following partial derivatives are equal to zero for a network polynomial f:
(a) $\dfrac{\partial^2 f}{\partial \lambda_x \, \partial \lambda_{x'}}$, where x and x' are values (possibly equal) of the same variable.
(b) $\dfrac{\partial^2 f}{\partial \theta_{x|u} \, \partial \theta_{x'|u'}}$, where $x\mathbf{u}$ and $x'\mathbf{u}'$ are instantiations (possibly equal) of the same family.
What does this imply on the structure of the polynomial f?

Solution. Both vanish because every term of

$$f \;=\; \sum_{\mathbf{z}} \prod_{\theta_{x|u} \sim \mathbf{z}} \theta_{x|u} \prod_{\lambda_x \sim \mathbf{z}} \lambda_x$$

contains exactly one indicator of each variable X and exactly one parameter of each family $X\mathbf{U}$, each to the first power: a term is indexed by a complete instantiation $\mathbf{z}$, which assigns X a unique value and $X\mathbf{U}$ a unique instantiation.
(a) If $x \neq x'$, no term contains both λ_x and $\lambda_{x'}$, an instantiation compatible with x being incompatible with x'. So $\partial f / \partial \lambda_x$, got by deleting λ_x from the terms that contain it and discarding the others, mentions $\lambda_{x'}$ nowhere, and differentiating again gives 0. If $x = x'$, degree 1 makes $\partial f / \partial \lambda_x$ free of λ_x itself, so again the second derivative vanishes.
(b) Verbatim the same argument with $\theta_{x|u}$ and $\theta_{x'|u'}$ in place of the two indicators, a term being compatible with exactly one instantiation of the family $X\mathbf{U}$.
Structurally, then, f is linear not merely in each variable but in each *block* – the indicators Λ_X of a network variable, the parameters $\Theta_{X\mathbf{U}}$ of a family – with every term a product of exactly one variable from each of the $2n$ blocks. Hence

$$f \;=\; \sum_{x} \lambda_x \frac{\partial f}{\partial \lambda_x}, \qquad f \;=\; \sum_{x\mathbf{u}} \theta_{x|u} \frac{\partial f}{\partial \theta_{x|u}},$$

the first sum over the values of any one fixed variable and the second over the instantiations of any one fixed family.

Problem 12.5. Provide probabilistic semantics for the following second partial derivatives of the network polynomial:

$$\frac{\partial^2 f}{\partial \lambda_x \, \partial \lambda_y}(e), \qquad \frac{\partial^2 f}{\partial \theta_{x|u} \, \partial \lambda_y}(e), \qquad \frac{\partial^2 f}{\partial \theta_{x|u} \, \partial \theta_{y|v}}(e).$$

Provide a circuit propagation scheme that will compute these derivatives and discuss its time and space complexity.

Solution. (i) $\partial^2 f / \partial \lambda_x \partial \lambda_y(e) = \Pr(x, y, e - XY)$ when $X \neq Y$, and 0 when $X = Y$ by Exercise 12.4(a). Deleting λ_x and then λ_y from the terms of Definition 12.1, and using $\prod_{\theta \sim \mathbf{z}} \theta = \Pr(\mathbf{z})$ (chain rule),

$$\frac{\partial^2 f}{\partial\lambda_x\partial\lambda_y} = \sum_{\mathbf{z}\sim xy} \Pr(\mathbf{z}) \prod_{\lambda_w\sim\mathbf{z},\, W\notin\{X,Y\}} \lambda_w,$$

and at e the surviving $\mathbf{z}$ are exactly those compatible with xy and with $e - XY$: the probability of the evidence obtained by setting X to x and Y to y.
(ii) $\theta_{x|u}\,\partial^2 f/\partial\theta_{x|u}\,\partial\lambda_y(e) = \Pr(x, \mathbf{u}, y, e - Y)$. Degree 1 in $\theta_{x|u}$ gives

$$\theta_{x|u}\frac{\partial f}{\partial\theta_{x|u}} = \sum_{\mathbf{z}\sim x\mathbf{u}} \Pr(\mathbf{z}) \prod_{\lambda_w\sim\mathbf{z}} \lambda_w,$$

and differentiating by λ_y at e retains the $\mathbf{z}$ compatible with $x\mathbf{u}$, with y, and with $e - Y$; this is the family marginal of $X\mathbf{U}$ under e with Y reset to y, and it vanishes if y contradicts $x\mathbf{u}$.
(iii) $\theta_{x|u}\theta_{y|v}\,\partial^2 f/\partial\theta_{x|u}\,\partial\theta_{y|v}(e) = \Pr(x, \mathbf{u}, y, \mathbf{v}, e)$, the joint marginal of the two families under e, by the same computation applied twice – and 0 when the two families coincide, by Exercise 12.4(b).
A propagation scheme. All three are ordinary first derivatives evaluated at a modified indicator vector. For evidence e and an instantiation $\mathbf{s}$ of a set $\mathbf{S}$, let $\rho_{e,\mathbf{s}}$ be the indicator vector of the evidence "e and $\mathbf{s}$", that is $\rho_{e,\mathbf{s}}(\lambda_w) = 1$ iff $w \sim e$ and either $W \notin \mathbf{S}$ or $w \sim \mathbf{s}$. One evaluation-plus-differentiation pass of Algorithm 34 or 35 on the *same* circuit with that input costs $O(|AC|)$ and gives, by Theorem 12.2 applied to that evidence,

$$\frac{\partial f}{\partial\lambda_x}(\rho_{e,\mathbf{s}}) = \Pr\big(x,\ (e \wedge \mathbf{s}) - X\big),$$
$$\theta_{x|u}\frac{\partial f}{\partial\theta_{x|u}}(\rho_{e,\mathbf{s}}) = \Pr\big(x, \mathbf{u},\, e,\, \mathbf{s}\big).$$

So $\mathbf{s} = y$ yields, in one pass, every derivative of type (i) and of type (ii) for that fixed y, and $\mathbf{s} = y\mathbf{v}$ yields every derivative of type (iii) for that fixed $y\mathbf{v}$ – a whole row of the second-derivative matrix per pass. Derivatives covered by Exercise 12.4 are reported as 0 without computation.
Complexity. With m indicators and q parameters in all ($m \le q$, both $O(n\exp(w))$), types (i) and (ii) need m passes and type (iii) needs q, so the time is $O((m+q)|AC|) = O(q\,|AC|)$, against $O(q^2|AC|)$ for re-evaluating once per pair. Each pass recomputes from scratch, so the working space is the two register arrays, $O(|AC|)$, plus the output – $\Theta(m^2 + q^2)$ if all second derivatives are retained, $O(|AC|)$ for a single row.

Problem 12.6. Suppose we extend the notion of evidence to allow for constraining the value of a variable instead of fixing it. For example, if X is a variable with three values x_1, x_2, and x_3, then a piece of evidence on variable X may be $X = x_1 \vee X = x_2$, which rules out value x_3 without committing to either value x_1 or x_2. Given this extended notion of evidence, known as a finding, show how we can evaluate a network polynomial so that its value equals the probability of the given findings.

Solution. Evaluate the polynomial with each indicator set to

$$\lambda_x \;=\; \begin{cases} 1, & \text{if } x \in S_X, \\ 0, & \text{otherwise,} \end{cases}$$

where the finding on X is $\beta_X = \bigvee_{x\in S_X}(X = x)$ and a variable on which nothing is observed gets the trivial finding $S_X = \mathcal{D}_X$. Then $f(\beta) = \Pr(\beta)$ for $\beta = \bigwedge_{X\in\mathbf{Z}}\beta_X$.
Indeed, in each term of Definition 12.1 the θ-product is $\Pr(\mathbf{z})$ by the chain rule, and the λ-product contains exactly one indicator per variable (Exercise 12.4), namely $\lambda_{\mathbf{z}(X)}$, so

$$\prod_{\lambda_x\sim\mathbf{z}} \lambda_x = \prod_{X\in\mathbf{Z}} [\,\mathbf{z}(X) \in S_X\,] = \begin{cases} 1, & \text{if } \mathbf{z} \models \beta, \\ 0, & \text{otherwise,} \end{cases}$$

a product of zeros and ones being 1 exactly when $\mathbf{z}$ satisfies every finding. Hence $f(\beta) = \sum_{\mathbf{z}\models\beta}\Pr(\mathbf{z}) =$

$\Pr(\beta)$, the event β being the disjoint union of the instantiations satisfying it. Theorem 12.1 is the case $|S_X| = 1$ on the instantiated variables and $S_X = \mathcal{D}_X$ on the rest.
The circuit itself is untouched – only the leaf inputs differ – so a finding query still costs one bottom-up pass, $O(|AC|)$, and the same argument one differentiation deeper gives $\partial f/\partial\lambda_x(\beta) = \Pr(x, \beta - X)$, with $\beta - X$ the result of making the finding on X trivial, and $\theta_{x|u}\,\partial f/\partial\theta_{x|u}(\beta) = \Pr(x, \mathbf{u}, \beta)$.

Problem 12.7. Consider the construction of an arithmetic circuit using variable elimination (Section 12.4.1). For each addition node n constructed by this algorithm, let $\mathrm{var}(n)$ stand for the variable X whose elimination has led to the construction of node n. Given some MAP variables $\mathbf{M}$ and evidence e, show that the following procedure generates an upper bound on the MAP probability $\mathrm{MAP}_P(\mathbf{M}, e)$: evaluate the circuit while treating an addition node n as a maximization node if $\mathrm{var}(n) \in \mathbf{M}$.

Solution. The modified evaluation computes

$$V \;=\; \bigotimes_{X_n}\bigotimes_{X_{n-1}}\cdots\bigotimes_{X_1} F(\mathbf{z}), \qquad \bigotimes_{X_i} = \begin{cases} \max_{X_i}, & X_i \in \mathbf{M}, \\ \sum_{X_i}, & X_i \notin \mathbf{M}, \end{cases}$$

where $\pi = X_1, \ldots, X_n$ is the elimination order used to build the circuit (X_1 eliminated first, hence innermost) and $F(\mathbf{z}) = \Pr(\mathbf{z})\,[\mathbf{z} \sim e] \geq 0$ is the term of the network polynomial at evidence e. For the addition nodes created when X_k is eliminated are exactly those with $\mathrm{var}(n) = X_k$, and each adds the entries of the product factor over the values of X_k, so making them maximization nodes replaces $\sum_{X_k}$ by $\max_{X_k}$ and changes nothing else; and the correctness induction for variable elimination uses only that the operator commutes with multiplication by a factor not mentioning X_k, which holds for max as well since every quantity here is nonnegative ($\max_x c\,g(x) = c\max_x g(x)$ for $c \geq 0$).
Next, for nonnegative G,

$$\sum_y \max_x G(x, y) \;\geq\; \max_x \sum_y G(x, y),$$

since if x^* attains the right-hand maximum then $\max_x G(x, y) \geq G(x^*, y)$ for every y, and summing over y gives it. Both $\sum$ and max are monotone in their argument, so replacing a fragment $\sum_Y \max_X$ of the expression for V by $\max_X \sum_Y$ never increases the value of the whole. Bubble-sorting the operator sequence until every max precedes every $\sum$ – at most $\binom{n}{2}$ transpositions, operators of the same kind commuting freely – therefore produces a nonincreasing chain from V down to

$$\max_{\mathbf{M}} \sum_{\mathbf{Z}\setminus\mathbf{M}} F(\mathbf{z}) \;=\; \mathrm{MAP}_P(\mathbf{M}, e),$$

so $V \geq \mathrm{MAP}_P(\mathbf{M}, e)$, as required. The bound is exact when no transposition is needed, that is when π is $\mathbf{M}$-constrained.

11.2 Exercises 12.8–12.14

Problem 12.8. Given a Bayesian network and some evidence e, show how to construct a Boolean formula whose models are precisely the MPE instantiations of evidence e. The complexity of your algorithm should be $O(n \exp(w))$, where n is the network size and w is the width of a given elimination order.

Solution. Compile, maximize, and read the formula off the evaluated circuit. Assume $\mathrm{MPE}_P(e) > 0$; otherwise every e-compatible instantiation has probability zero and the notion degenerates (return the formula for "compatible with e", or $\bot$).
Run variable elimination along π keeping a trace of its operations (Section 12.4.1), giving an arithmetic circuit AC for the network polynomial, of size and construction time $O(n \exp(w))$. Replace every addition node by a maximization node, giving the maximizer circuit AC^m of Definition 12.4; set

$\lambda_x = 1$ iff $x \sim e$, set each $\theta_{x|u}$ to its network value, and evaluate bottom-up into registers $\mathrm{vr}(v)$, so that the root holds $AC^m(e) = \mathrm{MPE}_P(e)$ (Section 12.3.2).
Introduce one Boolean indicator I_x per value of each variable and define, bottom-up,

- $\alpha_v = I_x$ at a leaf λ_x, and $\alpha_v = \top$ at a leaf $\theta_{x|u}$;
- $\alpha_v = \alpha_{c_1} \wedge \cdots \wedge \alpha_{c_k}$ at a multiplication node;
- $\alpha_v = \bigvee_{i\,:\,\mathrm{vr}(c_i)=\mathrm{vr}(v)} \alpha_{c_i}$ at a maximization node, keeping only the children that attain the maximum.

The formula is $\Phi = \alpha_r \wedge \Delta$, with r the root and

$$\Delta \;=\; \bigwedge_X \Big[\bigvee_x I_x \;\wedge \bigwedge_{x \neq x'} (\neg I_x \vee \neg I_{x'}) \Big]$$

the usual indicator clauses (11.8)–(11.9). Kept as a DAG with one internal node per circuit node, $|\Phi| = O(|AC^m|) + O(\sum_X |X|^2) = O(n \exp(w))$, the second term because $\sum_X |X| \leq n$ while $|X| \leq \exp(w)$ for every X, each X occurring in some cluster of the order.
Correctness. Complete subcircuits (Definition 12.5) of AC^m correspond one to one with the terms of the network polynomial, hence with the network instantiations z, the subcircuit for z containing exactly one leaf λ_x per variable, the one with $x \sim z$. Call a complete subcircuit *maximal* if every maximization node in it chooses a child c with $\mathrm{vr}(c) = \mathrm{vr}(v)$.
(i) A maximal complete subcircuit has value $\mathrm{vr}(v)$ at each of its nodes, by induction: a multiplication node keeps all children, so its value is $\prod_c \mathrm{vr}(c) = \mathrm{vr}(v)$, and a maximization node keeps one child of full value. At the root the value is $\mathrm{MPE}_P(e)$.
(ii) Conversely $\mathrm{val}(v) \leq \mathrm{vr}(v)$ always, values being nonnegative and a maximization node picking one child; and equality pushes downward from $\mathrm{val}(r) = \mathrm{vr}(r) > 0$: at a multiplication node $\prod_c \mathrm{val}(c) = \prod_c \mathrm{vr}(c)$ with all factors positive forces $\mathrm{val}(c) = \mathrm{vr}(c)$ childwise, and at a maximization node with chosen child c, $\mathrm{vr}(v) = \mathrm{val}(c) \leq \mathrm{vr}(c) \leq \mathrm{vr}(v)$.
So the maximal complete subcircuits are exactly those of maximal value, i.e. exactly the MPE instantiations of e (the recovery procedure of Section 12.3.2). Finally Δ forces every model of Φ to be ω_z for a unique instantiation z, and by construction α_r holds under ω_z iff some maximal complete subcircuit has all its λ-leaves compatible with z – one per variable – that is, iff z is an MPE instantiation. Hence

$$\mathrm{Models}(\Phi) \;=\; \{\, \omega_z \;:\; z \text{ is an MPE instantiation of } e \,\}.$$

Compilation, conversion, evaluation and the construction of the α_v are each linear in the circuit size, so the whole algorithm is $O(n \exp(w))$.
Unlike in Exercise 11.10, the clauses (11.9) cannot be dropped here: α_r is monotone, so any assignment setting more indicators true would satisfy it too, admitting models that select two values of one variable and describe no instantiation at all.

Problem 12.9. Provide an algorithm for generating circuits that can be used to compute MAP. Describe the computational complexity of the method.

Solution. Compile with an $\mathbf{M}$-constrained elimination order and turn the addition nodes of the MAP variables into maximization nodes.

1. Choose an order π in which every variable of $\mathbf{Y} = \mathbf{Z} \setminus \mathbf{M}$ precedes every variable of $\mathbf{M}$ (run min-fill on $\mathbf{Y}$ first, then on $\mathbf{M}$); let $w_{\mathbf{M}}$, the constrained width, be its width.
2. Run the circuit-generating variable elimination of Section 12.4.1 along π, labelling every addition node with the variable $\mathrm{var}(n)$ whose elimination created it, as in Exercise 12.7.
3. Replace each addition node with $\mathrm{var}(n) \in \mathbf{M}$ by a maximization node, leaving the others alone; call the result AC^{map}.

Evaluating AC^{map} bottom-up with $\lambda_x = 1$ iff $x \sim e$ leaves

$$\mathrm{MAP}_P(\mathbf{M}, e) \;=\; \max_{\mathbf{m}} \sum_{\mathbf{y}} \Pr(\mathbf{m}, \mathbf{y}, e)$$

at the root, and recording the maximizing child at each maximization node reads off a MAP instanti-

ation directly – the node created by eliminating M has one child per value of M.
The constrained order is what makes this exact rather than an upper bound: a circuit node is created only after its children, so no node created while eliminating some $M \in \mathbf{M}$ is a descendant of one created while eliminating some $Y \in \mathbf{Y}$; hence in AC^{map} no maximization node lies below an addition node, and bottom-up evaluation performs $\max_{\mathbf{m}} \sum_{\mathbf{y}}$, never $\sum_{\mathbf{y}} \max_{\mathbf{m}}$. Formally this is the variable-elimination induction with max in place of $\sum$ for the $\mathbf{M}$ variables, licensed by nonnegativity as in Exercise 12.7.
Complexity: $O(n \exp(w_{\mathbf{M}}))$ to construct the circuit, with the conversion, evaluation and extraction all linear in its size, so each later evidence costs one pass. Here $w_{\mathbf{M}} \geq w$ and the gap can be arbitrarily large – unavoidably, since D-MAP is NP-complete already on polytrees of treewidth at most 2 (Theorem 11.6), while $O(n \exp(w))$ would place MAP alongside MPE. Compiling with an unconstrained order and maximizing the same nodes costs only $O(n \exp(w))$ but returns the upper bound of Exercise 12.7 instead. When $\mathbf{M}$ is all of $\mathbf{Z}$, every addition node becomes a maximization node and AC^{map} is the maximizer circuit of Definition 12.4.

Problem 12.10. Consider a Bayesian network with structure $A \to B \to C$, where all variables are binary and its jointree is $A - AB - BC$. Construct the arithmetic circuit embedded in this jointree as given in Section 12.4.2, assuming that BC is the root cluster and that CPTs and evidence indicators for variables A, B, and C are assigned to clusters A, AB, and BC, respectively.
That is: the jointree has three clusters $\mathbf{C}_1 = \{A\}$, $\mathbf{C}_2 = \{A, B\}$ and $\mathbf{C}_3 = \{B, C\}$, with edges $\mathbf{C}_1 - \mathbf{C}_2$ and $\mathbf{C}_2 - \mathbf{C}_3$, hence separators $\mathbf{S}_{12} = \{A\}$ and $\mathbf{S}_{23} = \{B\}$. Cluster BC is the root, so AB is its child and A is the child of AB. The CPT Θ_A and indicator λ_A are assigned to cluster A; the CPT $\Theta_{B|A}$ and indicator λ_B to cluster AB; the CPT $\Theta_{C|B}$ and indicator λ_C to cluster BC. Variables A, B, C have values a_1, a_2; b_1, b_2; c_1, c_2.

Solution. By Definition 12.7 with $\mathbf{C}_3 = BC$ as root, the parent-to-child chain is

$$\mathbf{C}_3 \longrightarrow \mathbf{S}_{23} \longrightarrow \mathbf{C}_2 \longrightarrow \mathbf{S}_{12} \longrightarrow \mathbf{C}_1,$$

so separator B is the child of cluster BC and the parent of cluster AB, and separator A is the child of AB and the parent of cluster A. The circuit therefore has one multiplication node per cluster instantiation (ten), one addition node per separator instantiation (four), the output addition node f, and the six indicator and ten parameter inputs. A cluster node multiplies the inputs assigned to its cluster that are compatible with its instantiation together with the compatible addition nodes of its child separators; a separator node adds the compatible multiplication nodes of its child cluster; f adds the multiplication nodes of the root cluster. Bottom-up:

$$\begin{aligned}
n_{a_1} &= \lambda_{a_1} \star \theta_{a_1}, & n_{a_2} &= \lambda_{a_2} \star \theta_{a_2},\\
s_{a_1} &= n_{a_1}, & s_{a_2} &= n_{a_2},\\
n_{a_1b_1} &= \lambda_{b_1} \star \theta_{b_1|a_1} \star s_{a_1}, & n_{a_1b_2} &= \lambda_{b_2} \star \theta_{b_2|a_1} \star s_{a_1},\\
n_{a_2b_1} &= \lambda_{b_1} \star \theta_{b_1|a_2} \star s_{a_2}, & n_{a_2b_2} &= \lambda_{b_2} \star \theta_{b_2|a_2} \star s_{a_2},\\
s_{b_1} &= n_{a_1b_1} + n_{a_2b_1}, & s_{b_2} &= n_{a_1b_2} + n_{a_2b_2},\\
n_{b_1c_1} &= \lambda_{c_1} \star \theta_{c_1|b_1} \star s_{b_1}, & n_{b_1c_2} &= \lambda_{c_2} \star \theta_{c_2|b_1} \star s_{b_1},\\
n_{b_2c_1} &= \lambda_{c_1} \star \theta_{c_1|b_2} \star s_{b_2}, & n_{b_2c_2} &= \lambda_{c_2} \star \theta_{c_2|b_2} \star s_{b_2},
\end{aligned}$$

and the output node

$$f = n_{b_1c_1} + n_{b_1c_2} + n_{b_2c_1} + n_{b_2c_2},$$

the n-nodes being multiplication nodes and the s-nodes and f addition nodes. Unfolding from the root gives $\sum_{i,j,k} \lambda_{a_i}\lambda_{b_j}\lambda_{c_k}\,\theta_{a_i}\theta_{b_j|a_i}\theta_{c_k|b_j}$ (Check!), the network polynomial of the chain, as Theorem 12.5 guarantees; the counts match its bound, $3 \cdot 2^2 = 12 \geq 10$ multiplication nodes and $2 \cdot 2^1 = 4$ separator addition nodes. The nodes s_{a_i} have a single child each, cluster A coinciding with its parent separator, but the uncollapsed form is the one Definition 12.7 prescribes and the one that preserves the strict alternation of Exercise 12.11.

Problem 12.11. Show that the arithmetic circuit embedded in a jointree has the following properties:

- (a) The circuit alternates between addition and multiplication nodes.
- (b) Each multiplication node has a single parent.

(Recall Definition 12.7: given a rooted jointree with an assignment of CPTs and evidence indicators to clusters, the embedded circuit has one output addition node f; one addition node s per instantiation $\mathbf{s}$ of each separator $\mathbf{S}$; one multiplication node c per instantiation $\mathbf{c}$ of each cluster $\mathbf{C}$; and input nodes λ_x and $\theta_{x|\mathbf{u}}$. The children of f are the multiplication nodes of the root cluster; the children of s are the compatible multiplication nodes of the child cluster of $\mathbf{S}$; the children of c are the compatible addition nodes of the child separators of $\mathbf{C}$ together with the compatible inputs $\theta_{x|\mathbf{u}}$ and λ_x whose CPT $\Theta_{X|\mathbf{U}}$ or indicator λ_X is assigned to $\mathbf{C}$.)

Solution. Both properties are read off the children in Definition 12.7, using that in a rooted jointree every nonroot cluster has exactly one parent separator and every separator exactly one parent and one child cluster.

(a) An addition node is either the output node f, whose children are the multiplication nodes of the root cluster, or a separator node s, whose children are the multiplication nodes c with $\mathbf{c} \sim \mathbf{s}$ of the child cluster of $\mathbf{S}$; either way every child is a multiplication node, so an addition node has neither an addition child nor a leaf child. A multiplication node c has as children the addition nodes s with $\mathbf{s} \sim \mathbf{c}$ of the child separators of $\mathbf{C}$, together with input nodes, so it has no multiplication child. Hence the types strictly alternate $+, \star, +, \star, \ldots$ down every path from the root, one multiplication level per cluster and one addition level per separator, and leaves have only multiplication parents – exactly the structural assumption of Algorithm 35.

(b) All parents of a multiplication node c are addition nodes by (a), so it is enough to see which addition nodes list c as a child.

(i) $\mathbf{C}$ is the root cluster. It is the child of no separator, and separator nodes take their children from their child cluster, so no separator node lists c; the output node f is its only parent.

(ii) $\mathbf{C}$ is not the root cluster. Then f, which takes children only from the root cluster, does not list c, and the unique parent separator $\mathbf{S}$ of $\mathbf{C}$ is the only separator whose child cluster is $\mathbf{C}$. Its node s lists c precisely when $\mathbf{s} \sim \mathbf{c}$, and since $\mathbf{S} \subseteq \mathbf{C}$ exactly one instantiation of $\mathbf{S}$ is compatible with $\mathbf{c}$, namely the projection $\mathbf{c}|_{\mathbf{S}}$. So c again has exactly one parent. ■

Problem 12.12. Given an arithmetic circuit that is embedded in a binary jointree, describe a circuit propagation scheme that will evaluate and differentiate the circuit under the following constraints:

- The method can use only two registers for each addition or leaf node but no registers for multiplication nodes.
- The time complexity of the algorithm is linear in the circuit size.

Compare your developed scheme to the Shenoy-Shafer architecture for jointree propagation. Recall that a binary jointree is one in which each cluster has at most three neighbors.

Solution. Keep two registers at every addition node and every leaf and none at multiplication nodes. Exercise 12.11 makes that enough: a multiplication node c has a unique parent p, an addition node, so the chain rule gives

$$\mathrm{dr}(c) \;=\; \mathrm{dr}(p) \cdot \frac{\partial\, \mathrm{vr}(p)}{\partial\, \mathrm{vr}(c)} \;=\; \mathrm{dr}(p),$$

and all children of c are addition nodes or leaves, which do have value registers, so $\mathrm{vr}(c) = \prod_{v \in \mathrm{ch}(c)} \mathrm{vr}(v)$ is recomputable on demand.

Upward pass. Set $\mathrm{vr}(\lambda_x) = 1$ if $x \sim e$ and 0 otherwise, and each $\mathrm{vr}(\theta_{x|\mathbf{u}})$ to its network parameter; then visit addition nodes children-before-parents,

$$\mathrm{vr}(p) \;\longleftarrow\; \sum_{c \,\in\, \mathrm{ch}(p)} \; \prod_{v \,\in\, \mathrm{ch}(c)} \mathrm{vr}(v),$$

every v on the right being an addition node or a leaf, hence already filled; $\mathrm{vr}(c)$ is formed and consumed inside the sum and never stored.
Downward pass. Set $\mathrm{dr}(r) = 1$ at the output node and 0 at every other addition node and leaf; then visit addition nodes parents-before-children, executing for each multiplication child c of p

$$\mathrm{dr}(v) \;\longleftarrow\; \mathrm{dr}(v) \;+\; \mathrm{dr}(p) \prod_{v' \in \mathrm{ch}(c),\, v' \neq v} \mathrm{vr}(v'), \qquad v \in \mathrm{ch}(c),$$

which is line 10 of Algorithm 34 with $\mathrm{dr}(c)$ replaced by $\mathrm{dr}(p)$.
Correctness. The upward pass merely inlines the multiplication nodes, so it computes the values of line 2 of Algorithm 34. The downward pass accumulates $\mathrm{dr}(v) = \sum_{c \in \mathrm{pa}(v)} \mathrm{dr}(c) \prod_{v' \neq v} \mathrm{vr}(v')$, the identity of Section 12.3.1 for a node all of whose parents are multiplication nodes – which by Exercise 12.11(a) every addition node and every leaf is, so line 8 of Algorithm 34 never fires for such a v and no addition-parent term is missing. The top-down order makes $\mathrm{dr}(p)$ final before it is used. Hence $\mathrm{dr}(v) = \partial f / \partial\, \mathrm{vr}(v)$ at every leaf, and Theorem 12.2 applies unchanged.
In a binary jointree a cluster has at most three neighbours, hence at most two child separators, so $k = |\mathrm{ch}(c)|$ is $O(1)$ and the k sibling products may simply be formed one at a time: the upward pass costs $\sum_c O(k)$ and the downward $\sum_c O(k^2)$, both linear in the circuit size, with no division and no special handling of zeros. (Without the binary assumption, prefix and suffix products $L_i = \prod_{j<i} \mathrm{vr}(v_j)$, $R_i = \prod_{j>i} \mathrm{vr}(v_j)$ give $O(k)$ using $O(k)$ scratch words reused from node to node.)
The scheme is Shenoy-Shafer. That architecture stores two factors per separator, the messages M_{ij} and M_{ji}, and nothing at the clusters; here $\mathrm{vr}(s)$ is the entry of the inward message at $\mathbf{s}$ and $\mathrm{dr}(s)$ the entry of the outward one, with nothing at the cluster-instantiation nodes – space exponential in separator size, not cluster size. Its message $M_{ij} = \sum_{\mathbf{C}_i \setminus \mathbf{S}_{ij}} \Phi_i \prod_{k \neq j} M_{ki}$ (7.4) re-multiplies assigned factors and incoming messages instead of materializing a cluster table, which is what recomputing $\mathrm{vr}(c)$ and the sibling products does; and its cluster marginal $\Pr(\mathbf{C}_i, e) = \Phi_i \prod_k M_{ki}$ (7.5) is

$$\mathrm{dr}(c)\, \mathrm{vr}(c) \;=\; \mathrm{dr}(p) \prod_{v \in \mathrm{ch}(c)} \mathrm{vr}(v),$$

the very expression the downward pass forms. The binary restriction plays the same role in both: in Section 7.7.3 it keeps $\alpha = \sum_i n_i^2$ linear in n, here it bounds the children of a multiplication node. Only the bookkeeping differs – a cluster waiting for messages from all neighbours but j becomes a static topological traversal fixed when the circuit is compiled.

Problem 12.13. Given an arithmetic circuit that is embedded in a jointree, describe a circuit propagation scheme that will evaluate and differentiate the circuit under the following constraints:

- The method can use only one register $\mathrm{dvr}(v)$ for each circuit node v.
- When the algorithm terminates, the register $\mathrm{dvr}(v)$ contains the product $\mathrm{dr}(v)\, \mathrm{vr}(v)$, where $\mathrm{dr}(v)$ and $\mathrm{vr}(v)$ are as computed by Algorithm 34.
- The time complexity of the algorithm is linear in the circuit size.

Compare your developed scheme to the Hugin architecture for jointree propagation.
(Algorithm 34 is the two-pass circuit propagation scheme: an upward pass computes the value $\mathrm{vr}(v)$ of every node at the given evidence, and a downward pass computes $\mathrm{dr}(v) = \partial\, \mathrm{vr}(r) / \partial\, \mathrm{vr}(v)$ for the circuit root r, using

$$\mathrm{dr}(v) \leftarrow \mathrm{dr}(v) + \mathrm{dr}(p)$$

for an addition parent p and

$$\mathrm{dr}(v) \leftarrow \mathrm{dr}(v) + \mathrm{dr}(p) \prod_{v' \neq v} \mathrm{vr}(v')$$

for a multiplication parent p.)

Solution. One register suffices because $\mathrm{dr}(v)\mathrm{vr}(v)$ obeys a pure *sum* recurrence downward. By Exercise 12.11 the circuit strictly alternates, every multiplication node has a single (addition) parent, and every

parent of an addition node or leaf is a multiplication node; so the circuit is layered (layer 0 the root) with all parents of a node in one layer. Algorithm 34 gives $\mathrm{dr}(c) = \mathrm{dr}(p)$ at a multiplication node c with its unique addition parent p, hence for $\mathrm{vr}(p) \neq 0$

$$\text{(I)} \quad \mathrm{dr}(c)\,\mathrm{vr}(c) \;=\; \frac{\mathrm{dr}(p)\,\mathrm{vr}(p)}{\mathrm{vr}(p)}\;\mathrm{vr}(c),$$

while at an addition node or leaf v, all of whose parents are multiplication nodes,

$$\begin{aligned}\text{(II)} \quad \mathrm{dr}(v)\,\mathrm{vr}(v) &= \mathrm{vr}(v)\sum_{c\in\mathrm{pa}(v)}\mathrm{dr}(c)\prod_{v'\in\mathrm{ch}(c),\,v'\neq v}\mathrm{vr}(v') \\ &= \sum_{c\in\mathrm{pa}(v)}\mathrm{dr}(c)\prod_{v'\in\mathrm{ch}(c)}\mathrm{vr}(v') \;=\; \sum_{c\in\mathrm{pa}(v)}\mathrm{dr}(c)\,\mathrm{vr}(c),\end{aligned}$$

so no sibling values are needed. Pass 1 is the upward pass of Algorithm 34 run in dvr, leaving $\mathrm{dvr}(v) = \mathrm{vr}(v)$ and the root at $\Pr(e)$. Pass 2 runs layer by layer under the invariant that on entering layer d its registers hold dr vr and all deeper layers still hold vr (true at layer 0, where $\mathrm{dr}(r) = 1$):

- (i) *Addition layer d.* For each addition node p set $t \leftarrow \sum_{c\in\mathrm{ch}(p)}\mathrm{dvr}(c) = \mathrm{vr}(p)$, then $\rho \leftarrow \mathrm{dvr}(p)/t$ if $t \neq 0$ and $\rho \leftarrow 0$ otherwise, and $\mathrm{dvr}(c) \leftarrow \rho\,\mathrm{dvr}(c)$ at every child: this is (I), and since circuit values are nonnegative $t = 0$ forces every $\mathrm{vr}(c) = 0$, so $\rho = 0$ is correct and no division by zero occurs.
- (ii) *Multiplication layer d.* Zero layer $d+1$, then $\mathrm{dvr}(v) \leftarrow \mathrm{dvr}(v) + \mathrm{dvr}(c)$ for each child v of each c: this is (II).

Each node collects all contributions before its own layer is processed, so the invariant propagates and every register terminates at $\mathrm{dr}(v)\mathrm{vr}(v)$; overwriting vr at layer $d+1$ is safe because the only later read of a value register is the t of layer $d+1$, which reads layer $d+2$. Each edge is traversed twice per pass, so the work is linear for an *arbitrary* jointree — no binary restriction as in Exercise 12.12.

Comparison with Hugin. This is Hugin propagation written node by node: one register per multiplication node (cluster instantiation) and one per addition node (separator instantiation) mirror Hugin's single Φ_i per cluster and Φ_{ij} per separator with no message tables, and by Theorem 12.2 they terminate at $\Pr(\mathbf{c}, e)$ and $\Pr(\mathbf{s}, e)$ — entry for entry Hugin's Φ_i and Φ_{ij}. Bullet (ii) is Hugin's marginalization $\Phi_{ij} \leftarrow \sum_{\mathbf{C}_i\setminus\mathbf{S}_{ij}}\Phi_i$ and bullet (i) its division-and-multiply $\Phi_j \leftarrow \Phi_j\Phi_{ij}/\Phi_{ij}^{\mathrm{old}}$, with $\rho \leftarrow 0$ at $t = 0$ the convention $f_1/f_2 = 0$ of Definition 7.7.

Problem 12.14. Consider the arithmetic circuit in Figure 12.13. Evaluate and differentiate this circuit at evidence $e = b_2$ using the propagation scheme given by Algorithm 36. Compute all MPE instantiations given the evidence and describe the meaning of derivatives with respect to inputs λ_{a_1} and $\theta_{c_2|a_2}$.

The circuit of Figure 12.13 is the arithmetic circuit of the Bayesian network with structure $A \to B$, $A \to C$, all three variables having two values $(a_1, a_2;\ b_1, b_2;\ c_1, c_2)$, and with parameters

θ_{a_1}	θ_{a_2}
.1	.9

$\theta_{b_1\|a_1}$	$\theta_{b_2\|a_1}$	$\theta_{b_1\|a_2}$	$\theta_{b_2\|a_2}$
.2	.8	.3	.7

$\theta_{c_1\|a_1}$	$\theta_{c_2\|a_1}$	$\theta_{c_1\|a_2}$	$\theta_{c_2\|a_2}$
.4	.6	.01	.99

Its structure, from the leaves upward, is the following (each node listed with its children; $\star$ denotes a multiplication node):

- $p_a = \lambda_a \star \theta_a$, for $a \in \{a_1, a_2\}$;
- $q_{c,a} = \lambda_c \star \theta_{c|a}$, for each of the four pairs c, a;
- $s_a = q_{c_1,a} + q_{c_2,a}$, an addition node, for $a \in \{a_1, a_2\}$;
- $h_{b,a} = \theta_{b|a} \star p_a$, for each of the four pairs b, a;
- $m_{b,a} = h_{b,a} \star s_a$, for each of the four pairs b, a;
- $t_b = m_{b,a_1} + m_{b,a_2}$, an addition node, for $b \in \{b_1, b_2\}$;

- $u_b = \lambda_b \star t_b$, for $b \in \{b_1, b_2\}$;
- the output addition node $f = u_{b_1} + u_{b_2}$.

Algorithm 36 is the maximizer-circuit propagation scheme: the addition nodes of the circuit are replaced by maximization nodes, an upward pass fills the value registers $\mathrm{vr}(v)$, then dr is set to 1 at the root and 0 elsewhere, and a downward pass sets

$$\mathrm{dr}(v) \leftarrow \max(\mathrm{dr}(v), \mathrm{dr}(p))$$

for each maximization parent p of v and

$$\mathrm{dr}(v) \leftarrow \max\Big(\mathrm{dr}(v),\ \mathrm{dr}(p) \prod_{v' \neq v} \mathrm{vr}(v')\Big)$$

for each multiplication parent p.

Solution. Replacing the addition nodes s_a, t_b, f by maximization nodes gives the maximizer circuit AC^m of Definition 12.4; at $e = b_2$ the indicators are $\lambda_{b_1} = 0$ and all others 1, with each θ leaf at its network value.

Upward pass.

$$\begin{aligned}
\mathrm{vr}(p_{a_1}) &= .1, & \mathrm{vr}(p_{a_2}) &= .9,\\
\mathrm{vr}(q_{c_1,a_1}) &= .4,\ \mathrm{vr}(q_{c_2,a_1}) = .6, & \mathrm{vr}(q_{c_1,a_2}) &= .01,\ \mathrm{vr}(q_{c_2,a_2}) = .99,\\
\mathrm{vr}(s_{a_1}) &= \max(.4, .6) = .6, & \mathrm{vr}(s_{a_2}) &= \max(.01, .99) = .99,\\
\mathrm{vr}(h_{b_1,a_1}) &= .02,\ \mathrm{vr}(h_{b_2,a_1}) = .08, & \mathrm{vr}(h_{b_1,a_2}) &= .27,\ \mathrm{vr}(h_{b_2,a_2}) = .63,\\
\mathrm{vr}(m_{b_1,a_1}) &= .012,\ \mathrm{vr}(m_{b_2,a_1}) = .048, & \mathrm{vr}(m_{b_1,a_2}) &= .2673,\ \mathrm{vr}(m_{b_2,a_2}) = .6237,\\
\mathrm{vr}(t_{b_1}) &= .2673, & \mathrm{vr}(t_{b_2}) &= .6237,\\
\mathrm{vr}(u_{b_1}) &= 0, & \mathrm{vr}(u_{b_2}) &= .6237,
\end{aligned}$$

so $AC^m(e) = \mathrm{vr}(f) = \max(0, .6237) = .6237 = \mathrm{MPE}_P(b_2)$.

Downward pass. Set $\mathrm{dr}(f) = 1$ and all else 0; the maximization node f passes 1 to both u_b, and Algorithm 36 then gives

$$\begin{aligned}
\mathrm{dr}(\lambda_{b_1}) &= .2673, & \mathrm{dr}(t_{b_1}) &= 0,\\
\mathrm{dr}(\lambda_{b_2}) &= .6237, & \mathrm{dr}(t_{b_2}) &= 1,\\
\mathrm{dr}(m_{b_2,a}) &= 1, & \mathrm{dr}(m_{b_1,a}) &= 0,\\
\mathrm{dr}(h_{b_2,a_1}) &= \mathrm{vr}(s_{a_1}) = .6, & \mathrm{dr}(h_{b_2,a_2}) &= \mathrm{vr}(s_{a_2}) = .99,\\
\mathrm{dr}(s_{a_1}) &= \mathrm{vr}(h_{b_2,a_1}) = .08, & \mathrm{dr}(s_{a_2}) &= \mathrm{vr}(h_{b_2,a_2}) = .63,\\
\mathrm{dr}(p_{a_1}) &= \max(0 \cdot .2,\ .6 \cdot .8) = .48, & \mathrm{dr}(p_{a_2}) &= \max(0 \cdot .3,\ .99 \cdot .7) = .693,
\end{aligned}$$

every contribution through $m_{b_1,a}$ being 0 and losing its maximization, so $\mathrm{dr}(h_{b_1,a}) = 0$; the s_a pass their registers down unchanged, giving $\mathrm{dr}(q_{c,a_1}) = .08$ and $\mathrm{dr}(q_{c,a_2}) = .63$. At the shared leaves λ_c the maximum over the two parents is taken: $\mathrm{dr}(\lambda_{c_1}) = \max(.032, .0063) = .032$ and $\mathrm{dr}(\lambda_{c_2}) = \max(.048, .6237) = .6237$. Collecting the leaf derivatives:

leaf	λ_{a_1}	λ_{a_2}	λ_{b_1}	λ_{b_2}	λ_{c_1}	λ_{c_2}
dr	.048	.6237	.2673	.6237	.032	.6237

leaf	θ_{a_1}	θ_{a_2}	$\theta_{b_1\|a_1}$	$\theta_{b_2\|a_1}$	$\theta_{b_1\|a_2}$	$\theta_{b_2\|a_2}$
dr	.48	.693	0	.06	0	.891

leaf	$\theta_{c_1\|a_1}$	$\theta_{c_2\|a_1}$	$\theta_{c_1\|a_2}$	$\theta_{c_2\|a_2}$
dr	.08	.08	.63	.63

These agree with Theorem 12.4, $\partial f^m_x / \partial \lambda_x(e) = \mathrm{MPE}_P(x, e - X)$, and with (12.10), $\theta_{x|\mathbf{u}}\, \partial f^m_{x,\mathbf{u}} / \partial \theta_{x|\mathbf{u}}(e) = \mathrm{MPE}_P(e, x, \mathbf{u})$, read off the joint $\Pr(a, b, c) = \theta_a \theta_{b|a} \theta_{c|a}$:

instantiation	$a_1b_1c_1$	$a_1b_1c_2$	$a_1b_2c_1$	$a_1b_2c_2$	$a_2b_1c_1$	$a_2b_1c_2$	$a_2b_2c_1$	$a_2b_2c_2$
Pr	.008	.012	.032	.048	.0027	.2673	.0063	.6237

for instance $\mathrm{MPE}_P(a_1, b_2) = \max(.032, .048) = .048 = \mathrm{dr}(\lambda_{a_1})$ (Check! for the remaining entries).
All MPE instantiations. These are the complete subcircuits (Definition 12.5) whose every maximization choice attains the parent's value; descending from the root:

- (i) at f: $\mathrm{vr}(u_{b_1}) = 0 \neq .6237$, so only u_{b_2}, giving $B = b_2$;
- (ii) at t_{b_2}: $\mathrm{vr}(m_{b_2,a_1}) = .048 \neq .6237$, so only m_{b_2,a_2}, which via h_{b_2,a_2} contributes p_{a_2}, giving $A = a_2$;
- (iii) at s_{a_2}: $\mathrm{vr}(q_{c_1,a_2}) = .01 \neq .99$, so only q_{c_2,a_2}, giving $C = c_2$.

Every maximization node has a unique maximizing child, so $a_2 b_2 c_2$ is the one MPE instantiation, of probability .6237.
By (12.9), $\mathrm{dr}(\lambda_{a_1}) = \mathrm{MPE}_P(a_1, e - A) = \mathrm{MPE}_P(a_1, b_2)$: the MPE probability that strengthening the evidence to $a_1 b_2$ would give, read off without re-evaluating the circuit; as $.048 < .6237 = \mathrm{dr}(\lambda_{a_2})$, the criterion of Exercise 12.15 says a_1 occurs in no MPE instantiation. By (12.7), $\mathrm{dr}(\theta_{c_2|a_2})$ is the rate of change of the maximizer polynomial restricted to terms containing $\theta_{c_2|a_2}$ — the restriction being what makes it well defined, since for $\theta = \theta_{c_2|a_2}$ the unrestricted $f^m(b_2) = \max(.63\,\theta, .048)$ has a kink at $\theta = .048/.63$ while $f^m_{c_2,a_2} = .63\,\theta$ is linear — and by (12.10) its scaled value

$$\theta_{c_2|a_2} \frac{\partial f^m_{c_2,a_2}}{\partial \theta_{c_2|a_2}}(e) = .99 \cdot .63 = .6237 = \mathrm{MPE}_P(b_2, c_2, a_2)$$

equals $\mathrm{MPE}_P(b_2)$, so the family instantiation $c_2 a_2$ does occur in an MPE instantiation.

11.3 Exercises 12.15–12.20

Problem 12.15. Let X be a binary variable, with values x and $\bar{x}$, in a Bayesian network with maximizer polynomial f^m. Show that every MPE instantiation for evidence e includes $X = x$ if and only if

$$\frac{\partial f^m_x}{\partial \lambda_x}(e) \;>\; \frac{\partial f^m_{\bar{x}}}{\partial \lambda_{\bar{x}}}(e).$$

Here f^m_x denotes the restriction of the maximizer polynomial to x in the sense of Definition 12.6. Assume that variable X is not instantiated by the evidence e.

Solution. Since X is unset by e we have $e - X = e$, so Theorem 12.4 identifies the two derivatives with $d_x = \mathrm{MPE}_P(x, e)$ and $d_{\bar{x}} = \mathrm{MPE}_P(\bar{x}, e)$, and the claim reads: every MPE instantiation for e sets $X = x$ iff $d_x > d_{\bar{x}}$. Writing $z \sim e$ for compatibility of a complete instantiation, binary X partitions $\{z : z \sim e\}$ into the two nonempty sets $\{z \sim xe\}$ and $\{z \sim \bar{x}e\}$, so

$$\mathrm{MPE}_P(e) \;=\; \max(d_x, d_{\bar{x}}). \qquad (*)$$

(i) $\Leftarrow$. If $d_x > d_{\bar{x}}$ then $\mathrm{MPE}_P(e) = d_x$ by $(*)$, and an MPE instantiation $z^\star$ setting $X = \bar{x}$ would satisfy $z^\star \sim \bar{x}e$, whence $d_x = \Pr(z^\star) \leq d_{\bar{x}} < d_x$ — a contradiction.
(ii) $\Rightarrow$, by contraposition. If $d_x \leq d_{\bar{x}}$ then $\mathrm{MPE}_P(e) = d_{\bar{x}}$ by $(*)$, and the nonempty finite set $\{z \sim \bar{x}e\}$ contains some $z^\star$ attaining $d_{\bar{x}} = \mathrm{MPE}_P(e)$ — an MPE instantiation setting $X = \bar{x}$.

Problem 12.16. Show the MLF encoded by the propositional sentence

$$\Delta \;=\; A \wedge (B \Rightarrow D) \wedge (C \Rightarrow B).$$

Recall the encoding convention of Section 12.4.3: a propositional sentence Δ over variables $V_1, \ldots, V_n$ encodes a multi-linear function over corresponding MLF variables, where each model ω of Δ contributes one term, and MLF variable X appears in that term if and only if ω sets the propositional variable X to true. The MLF encoded by Δ is the sum of the terms contributed by all of its models.

Solution. The MLF encoded by Δ is

$$f \;=\; A \;+\; AD \;+\; ABD \;+\; ABCD.$$

Over $\{A, B, C, D\}$ the first conjunct forces A true, and splitting on B: with B false, $C \Rightarrow B$ forces C false and D is free; with B true, $B \Rightarrow D$ forces D true and C is free. So Δ has exactly four models, each contributing the product of the variables it sets true:

model	A	B	C	D	encoded term
ω_1	true	false	false	false	A
ω_2	true	false	false	true	AD
ω_3	true	true	false	true	ABD
ω_4	true	true	true	true	$ABCD$

Problem 12.17. Let Δ_f be a propositional sentence encoding an MLF f (each model of Δ_f encodes the term containing exactly those MLF variables X for which the model sets the propositional variable V_X to true, and f is the sum of these terms). Let Γ_f be an equivalent NNF circuit that satisfies decomposability, determinism, and smoothness. Show that the arithmetic circuit extracted from Γ_f as given in Section 12.4.3 is a representation of the MLF f. Recall the extraction rules: replace every and-node by a multiplication node, every or-node by an addition node, every negative literal $\neg V_X$ by the constant 1, and every positive literal V_X by the MLF variable X.

Solution. Structural induction on Γ_f. For a node N let Δ_N be the sentence it represents, $\text{vars}(N)$ the variables labelling the leaves below it, g_N the polynomial computed at the extracted node, and

$$h_N \stackrel{\text{def}}{=} \sum_{\substack{\omega \models \Delta_N \\ \omega \text{ over } \text{vars}(N)}} t(\omega), \qquad t(\omega) = \prod_{\substack{X \in \text{vars}(N) \\ \omega(V_X) = \text{true}}} X,$$

the MLF encoded by Δ_N read over $\text{vars}(N)$. Smoothness is taken relative to the full variable set of Δ_f — what Exercise 2.11 delivers, and what is needed, since Section 2.7 smoothness alone constrains only an or-node's children and would let the root omit a variable of f — so $h_{\text{root}} = f$, and it suffices to prove $g_N = h_N$.

- (i) *Leaves.* For $N = V_X$ the single model over $\{X\}$ sets V_X true, of term X, and extraction gives the leaf X; for $N = \neg V_X$ the single model sets V_X false, contributing the empty product 1, and extraction gives the constant 1 — a negative literal omits a variable from a term rather than killing the term.
- (ii) *And-nodes (decomposability).* $\Delta_N = \bigwedge_i \Delta_{C_i}$ and $\text{vars}(N) = \bigcup_i \text{vars}(C_i)$, the union disjoint, so every ω over $\text{vars}(N)$ factors uniquely as $(\omega_1, \ldots, \omega_k)$ with $\omega \models \Delta_N$ iff $\omega_i \models \Delta_{C_i}$ for all i (each Δ_{C_i} mentioning only $\text{vars}(C_i)$) and $t(\omega) = \prod_i t(\omega_i)$; distributing the product over the sums gives $h_N = \prod_i h_{C_i}$, and extraction made N a multiplication node, so $g_N = \prod_i g_{C_i} = h_N$ by the hypothesis.
- (iii) *Or-nodes (smoothness, determinism).* $\Delta_N = \bigvee_i \Delta_{C_i}$ with $\text{vars}(C_i) = \text{vars}(N)$ by smoothness, so the models of each Δ_{C_i} are already models over $\text{vars}(N)$ and no term expansion is needed, and the models of Δ_N over $\text{vars}(N)$ are their union — disjoint by determinism ($\Delta_{C_i} \wedge \Delta_{C_j}$ inconsistent for $i \neq j$, which is also what keeps h_N an MLF rather than giving a twice-counted model coefficient 2). So the term sums add, $h_N = \sum_i h_{C_i}$, and extraction made N an addition node, giving $g_N = \sum_i g_{C_i} = h_N$.

At the root, $g_{\text{root}} = h_{\text{root}} = f$.

Problem 12.18. Prove Lemma 12.1 on Page 310, which states:
Lemma 12.1 (Path coefficients). Consider a maximizer circuit AC^m evaluated at some evidence e and let α be a path from the circuit root to some leaf node n. Define the coefficient r of path α as the product of values attained by nodes c, where c is not on path α but has a multiplication parent on α. Then $r \cdot k$ is the maximum value attained by any complete subcircuit that includes path α, where k is the value of leaf node n.
Here the value of a node is the one computed by the bottom-up evaluation pass of Algorithm 36 at

evidence e (a maximization node takes the maximum of its children's values, a multiplication node takes their product, a leaf takes its input value), and a complete subcircuit is as in Definition 12.5: start at the root, include all children of every visited multiplication node and exactly one child of every visited maximization node; its value is the product of the values of its leaves.

Solution. Induct on the length m of α, all values being nonnegative — leaves are indicators in $\{0,1\}$ or parameters in $[0,1]$, and max and product preserve nonnegativity — which is what licenses maximizing a product factor by factor. Read paths and complete subcircuits in the tree unwinding of the DAG, write $\mathrm{val}(S)$ for the product of S's leaf values with traversal multiplicity, and note that the nodes off $\alpha = v_0 \to \cdots \to v_m$ with a multiplication parent on α are exactly the siblings along α.
(a) $\mathrm{vr}(v) = \max\{\mathrm{val}(S) : S \text{ rooted at } v\}$ for every v, by bottom-up induction: at a leaf the only such S is v; at a maximization node the subcircuits are v plus one rooted at a single child, of maximum $\max_t \mathrm{vr}(c_t) = \mathrm{vr}(v)$; at a multiplication node with children $c_1, \ldots, c_j$ the choices at the children are independent and $\mathrm{val}(S) = \prod_t \mathrm{val}(S_t)$, so by nonnegativity ($\max_{a\in A, b\in B} ab = (\max A)(\max B)$)

$$\max_S \mathrm{val}(S) = \prod_{t=1}^{j} \max_{S_t} \mathrm{val}(S_t) = \prod_{t=1}^{j} \mathrm{vr}(c_t) = \mathrm{vr}(v).$$

(b) The lemma is the case $v =$ root of: for every node v and path α from v to a leaf n, with r the coefficient of α and $k = \mathrm{vr}(n)$,

$$\max\{\, \mathrm{val}(S) : S \text{ rooted at } v,\ S \supseteq \alpha \,\} \;=\; r \cdot k.$$

For $m = 0$ the only S is the leaf n, of value k, and r is the empty product 1. For $m > 0$ split $\alpha = v_0 \to \alpha'$ with α' running from the child v_1 to n and r' its coefficient, so by the hypothesis the best S' rooted at v_1 with $S' \supseteq \alpha'$ has value $r'k$.

- (i) v_0 a maximization node: $S \supseteq \alpha$ is v_0 plus such an S', with $\mathrm{val}(S) = \mathrm{val}(S')$, and v_0 contributes no siblings, so $r = r'$ and the maximum is rk.
- (ii) v_0 a multiplication node with children $v_1, c_2, \ldots, c_j$: S is v_0 plus an independent subcircuit at *each* child, constrained only by $S_1 \supseteq \alpha'$, so maximizing factor by factor via the hypothesis and (a),

$$\max_{S\supseteq\alpha} \mathrm{val}(S) \;=\; (r'k) \prod_{t=2}^{j} \mathrm{vr}(c_t) \;=\; rk,$$

the siblings along α being $c_2, \ldots, c_j$ together with those along α'.

Problem 12.19. Consider the network polynomial f for a Bayesian network and let $S = \{\theta_{x_1|u_1}, \ldots, \theta_{x_k|u_k}\}$ be a set of parameters with equal values in the same CPT; that is, all of them belong to the CPT $\Theta_{X|U}$ of a single variable X with parents U, and they share a common numeric value. Replace all these parameters in the polynomial f with a new variable η. What is the probabilistic meaning of $\partial f/\partial\eta(e)$ for a given evidence e?

Solution. Writing α for the event $\bigvee_{i=1}^{k}(X = x_i \wedge U = u_i)$,

$$\eta\,\frac{\partial f}{\partial \eta}(e) \;=\; \Pr(\alpha, e), \qquad \text{i.e.} \qquad \frac{\partial f}{\partial \eta}(e) \;=\; \frac{\Pr(\alpha, e)}{\eta} \quad (\eta \neq 0) :$$

the probability that e holds and the family XU takes one of the tied instantiations, divided by the shared value — the generalization of (12.4), which is the case $k = 1$.
By Definition 12.1 each term of f carries exactly one parameter of $\Theta_{X|U}$, namely $\theta_{x|u}$ for the restriction xu of its complete instantiation; hence no term contains two members of S, the substitution creates no η^2, and f stays multi-linear in η. Grouping the terms by which member of S they carry,

$$f \;=\; \sum_{i=1}^{k} \theta_{x_i|u_i}\, g_i \;+\; h, \qquad g_i \;=\; \frac{\partial f}{\partial \theta_{x_i|u_i}},$$

where h collects the terms carrying no member of S. No g_i contains a parameter of $\Theta_{X|U}$ and no term of h a member of S, so the substitution leaves $g_1, \ldots, g_k$ and h untouched, turns f into $\eta \sum_i g_i + h$, and gives the unconditional identity $\partial f/\partial \eta = \sum_i \partial f/\partial \theta_{x_i|u_i}$; multiplying by the common value $\eta = \theta_{x_i|u_i}$ and applying (12.4) of Theorem 12.2,

$$\eta \frac{\partial f}{\partial \eta}(e) \;=\; \sum_{i=1}^{k} \theta_{x_i|u_i} \frac{\partial f}{\partial \theta_{x_i|u_i}}(e) \;=\; \sum_{i=1}^{k} \Pr(x_i, u_i, e) \;=\; \Pr(\alpha, e),$$

the last step because $x_1u_1, \ldots, x_ku_k$ are distinct instantiations of the same variables XU, hence mutually exclusive.

Problem 12.20. Let f be an MLF and let f_m be another MLF obtained by including only the minimal terms of f. Here a term of f is minimal if the number of variables it contains is minimal among the terms of f. Suppose now that Δ_f is a CNF that encodes MLF f. Describe a procedure for obtaining an arithmetic circuit for MLF f_m. Hint: Consider Exercise 2.13.
For reference, Exercise 2.13 reads: Let Γ be an NNF circuit that satisfies decomposability, determinism, and smoothness. Consider the following procedure for generating a subcircuit Γ_m of circuit Γ. Assign an integer to each node of Γ as follows: an input node is assigned 0 if labelled with true or a positive literal, ∞ if labelled with false, and 1 if labelled with a negative literal; an or-node is assigned the minimum of the integers assigned to its children, and an and-node the sum of the integers assigned to its children. Obtain Γ_m from Γ by deleting every edge that extends from an or-node N to a child C where N and C have different integers assigned to them. Then the models of Γ_m are the minimum-cardinality models of Γ, where the cardinality of a model is the number of variables it sets to false.

Solution. Run Exercise 2.13 with the weights reversed — counting variables set to *true*, since under the encoding of Section 12.4.3 a term contains X exactly when the model sets V_X true, so term length is the true-count — and extract:

1. Compile Δ_f into an equivalent NNF circuit Γ_f satisfying decomposability, determinism, and smoothness (step 2 of Section 12.4.3, smoothed by Exercise 2.11 so that the root mentions every variable of Δ_f).
2. Label bottom-up with $i(\cdot)$: positive literal 1, negative literal 0, true 0, false ∞; an or-node the minimum of its children, an and-node the sum.
3. Delete every or-edge $N \to C$ with $i(C) \neq i(N)$, discard what is now unreachable from the root, and call the survivor Γ_m.
4. Extract the arithmetic circuit from Γ_m as in Section 12.4.3: and-node to multiplication, or-node to addition, $\neg V_X$ to the constant 1, V_X to X.

Steps 2–4 are three linear sweeps on a circuit no larger than Γ_f, so all the cost is in step 1. Correctness is three claims.
(i) For every node N,

$$i(N) \;=\; \min_{\substack{\omega \models \Delta_N \\ \omega \text{ over } \mathrm{vars}(N)}} \big|\{X \in \mathrm{vars}(N) : \omega(V_X) = \text{true}\}\big|,$$

with $\min \emptyset = \infty$: the length of the shortest term of the MLF encoded at N. This is the structural induction of Exercise 12.17 with min for $\sum$ — at an and-node decomposability makes the models a product over disjoint variable sets, so true-counts add and the minimum of the sum is the sum of the minima; at an or-node smoothness makes the children's models be models over the same vars(N), so the model sets union and the minima take a minimum.
(ii) The models of Γ_m are exactly the minimum-true-count models of Γ_f, i.e. those encoding the minimal terms of f. Each model ω has a unique certificate T_ω — descend from the root taking all children of an and-node and, by determinism, the unique child satisfied by ω at an or-node — carrying by decomposability and smoothness exactly one literal leaf per variable, namely ω's own, so ω's true-count is $w(T_\omega) = \sum_\ell i(\ell)$. An induction gives $w(T) \geq i(v)$ for every complete subcircuit T rooted at v, tight iff

T picks a child with $i(C) = i(N)$ at every or-node: at an and-node $w(T) = \sum_c w(T_c) \geq \sum_c i(c) = i(v)$, tight iff each part is; at an or-node with chosen child c, $w(T) = w(T_c) \geq i(c) \geq i(v)$, tight iff both inequalities are. Step 3 deletes exactly the non-tight or-edges, so T_ω survives iff $w(T_\omega) = i(\text{root})$; and as Γ_m only drops disjuncts, its models are those of Γ_f whose certificate survives. (If $i(\text{root}) = \infty$ then Δ_f is unsatisfiable and the construction returns the empty circuit.)

(iii) Γ_m inherits the three properties: no and-edge is deleted and at least one child of every or-node survives (the minimum is attained), so vars is unchanged at every surviving node, whence decomposability and smoothness; determinism holds because each surviving child's models are a subset of the original child's.

So Γ_m encodes f_m, and by Exercise 12.17 the circuit extracted at step 4 represents it.

12 Inference with Local Structure

Contents

12.1 Exercises 13.1–13.7 195
12.2 Exercises 13.8–13.14 202
12.3 Exercises 13.15–13.21 210
12.4 Exercises 13.22–13.22 217

12.1 Exercises 13.1–13.7

Problem 13.1. Consider the Bayesian network of Figure 13.2: it has root nodes $X_1, \ldots, X_n$ and a single leaf Y with edges $X_i \to Y$ for $i = 1, \ldots, n$, so Y has all of $X_1, \ldots, X_n$ as parents and the network has treewidth n. All variables take values in $\{0, 1\}$, each X_i has a uniform distribution ($\theta_{x_i=0} = \theta_{x_i=1} = 1/2$), and Y is the logical-or of its parents, so $\theta_{y|x_1,\ldots,x_n} = 1$ precisely when either $y = 1$ and $x_i = 1$ for some i, or $y = 0$ and $x_i = 0$ for all i. Under these assumptions the network polynomial factors as in Equation 13.1:

$$f = \left(\tfrac{1}{2}\right)^n \left(\lambda_{y=0} \prod_{i=1}^{n} \lambda_{x_i=0} \;+\; \lambda_{y=1} \sum_{i=1}^{n} \left(\prod_{j=1}^{i-1} \lambda_{x_j=0} \right) \lambda_{x_i=1} \left(\prod_{j=i+1}^{n} (\lambda_{x_j=0} + \lambda_{x_j=1}) \right) \right).$$

Provide an arithmetic circuit of size $O(n)$ for this polynomial.

Solution. Share one prefix chain and one suffix chain across the n terms of the inner sum, which a DAG does for free where computing each term's $\prod_{j<i} \lambda_{x_j=0}$ and $\prod_{j>i}(\lambda_{x_j=0} + \lambda_{x_j=1})$ separately would cost $\Theta(n^2)$ nodes. Take as leaves the $2n+2$ indicators and one constant $(1/2)^n$ (parameters substituted by value as in Figure 13.1(b), so the circuit is valid only at these values), and with $+$ and $\star$ the two node types of Definition 12.2 (a ternary node abbreviating two binary ones) set

$$\begin{aligned}
A_i &= +(\lambda_{x_i=0}, \lambda_{x_i=1}), && i = 1, \ldots, n,\\
P_0 = 1, \quad P_i &= \star(P_{i-1}, \lambda_{x_i=0}), && i = 1, \ldots, n,\\
S_{n+1} = 1, \quad S_i &= \star(A_i, S_{i+1}), && i = n, \ldots, 2,\\
T_i &= \star(P_{i-1}, \lambda_{x_i=1}, S_{i+1}), && i = 1, \ldots, n,\\
\Sigma &= +(T_1, +(T_2, \ldots)),\\
f &= \star\Big(\left(\tfrac{1}{2}\right)^n,\; +\big(\star(\lambda_{y=0}, P_n),\, \star(\lambda_{y=1}, \Sigma)\big)\Big).
\end{aligned}$$

Then $P_i = \prod_{j \le i} \lambda_{x_j=0}$ and $S_i = \prod_{j \ge i} A_j$, so T_i is exactly the i-th summand of Equation 13.1, Σ the inner sum, and the root f. Counting, with P_0, P_1, S_{n+1}, S_n costing no node and T_1, T_n one each,

$$\begin{aligned}
\text{leaves} &: 2n + 3,\\
\text{addition} &: n \text{ (the } A_i) + (n-1) \text{ } (\Sigma) + 1 \;=\; 2n,\\
\text{multiplication} &: (n-1) + (n-2) + (2n-2) + 3 \;=\; 4n - 2,
\end{aligned}$$

so at most $8n + 1$ nodes and, every internal node being binary, $O(n)$ edges.

Problem 13.2. Consider Exercise 13.1, that is, the network of Figure 13.2 with root nodes $X_1, \ldots, X_n$, leaf Y, edges $X_i \to Y$, uniform priors on the X_i, and Y the logical-or of its parents, whose polynomial is given by Equation 13.1:

$$f = \left(\tfrac{1}{2}\right)^n \left(\lambda_{y=0} \prod_{i=1}^{n} \lambda_{x_i=0} \;+\; \lambda_{y=1} \sum_{i=1}^{n} \left(\prod_{j=1}^{i-1} \lambda_{x_j=0} \right) \lambda_{x_i=1} \left(\prod_{j=i+1}^{n} (\lambda_{x_j=0} + \lambda_{x_j=1}) \right) \right).$$

Show that a smaller circuit can be constructed for this polynomial assuming that the circuit can contain subtraction nodes.

Solution. With $A_i = \lambda_{x_i=0} + \lambda_{x_i=1}$ and T_i the i-th summand of the inner sum, the whole sum collapses to a difference of two products,

$$\sum_{i=1}^{n} T_i \;=\; \prod_{i=1}^{n} A_i \;-\; \prod_{i=1}^{n} \lambda_{x_i=0},$$

which is the identity a subtraction node buys. Indeed, put

$$Q_k \;=\; \left(\prod_{j\le k}\lambda_{x_j=0}\right)\left(\prod_{j>k}A_j\right), \qquad k=0,1,\ldots,n,$$

so that $Q_0=\prod_i A_i$ and $Q_n=\prod_i \lambda_{x_i=0}$. For each i,

$$\begin{aligned} Q_{i-1}-Q_i &= \left(\prod_{j<i}\lambda_{x_j=0}\right)\big(A_i-\lambda_{x_i=0}\big)\left(\prod_{j>i}A_j\right)\\ &= \left(\prod_{j<i}\lambda_{x_j=0}\right)\lambda_{x_i=1}\left(\prod_{j>i}A_j\right) \;=\; T_i, \end{aligned}$$

because $A_i-\lambda_{x_i=0}=\lambda_{x_i=1}$, so summing over i telescopes to $\sum_i T_i = Q_0-Q_n$. Substituting into Equation 13.1,

$$f \;=\; \big(\tfrac{1}{2}\big)^n\big(\lambda_{y=0}\,\Pi_0 \;+\; \lambda_{y=1}\big(\Pi_A-\Pi_0\big)\big), \qquad \Pi_0=\prod_{i=1}^n\lambda_{x_i=0}, \quad \Pi_A=\prod_{i=1}^n A_i,$$

realized by n addition nodes $A_i=+(\lambda_{x_i=0},\lambda_{x_i=1})$, two chains of $n-1$ multiplication nodes for Π_A and Π_0, one subtraction node $\Delta=-(\Pi_A,\Pi_0)$, the two products $\star(\lambda_{y=0},\Pi_0)$, $\star(\lambda_{y=1},\Delta)$, one addition node combining them, and one multiplication by $(1/2)^n$: in all $2n+3$ leaves, $n+1$ addition, $2n+1$ multiplication and 1 subtraction node, so $5n+6$ against the $8n+1$ of Exercise 13.1 — subtraction expressing "at least one X_i is 1" by complementation where a monotone circuit must enumerate the n exclusive choices of the first parent that is 1.

Problem 13.3. Consider a Bayesian network consisting of binary nodes $X_1,\ldots,X_n,Y$ and edges $X_i\to Y$ for $i=1,\ldots,n$ (the structure of Figure 13.2, with all X_i roots and Y their common child). Assume that Y is true with probability ϵ_1 given that any of its parents are true, and that it is false with probability ϵ_0 given that all its parents are false; that is, Y is a soft-or of its parents. Provide an arithmetic circuit for this network of size $O(n)$ while making no assumptions about the distributions over nodes X_i.

Solution. The soft-or CPT takes only four distinct values however large n is, and that parameter equality is the local structure the circuit exploits: $\theta_{y|\mathbf{x}}$ is $\epsilon_0, 1-\epsilon_0$ for $y=0,1$ when every $x_i=0$, and $1-\epsilon_1,\epsilon_1$ for $y=0,1$ when some $x_i=1$. Writing the values of X_i as $0,1$, with arbitrary priors $\theta_{x_i=0},\theta_{x_i=1}$, abbreviate

$$\alpha_i=\lambda_{x_i=0}\,\theta_{x_i=0}, \qquad \beta_i=\lambda_{x_i=1}\,\theta_{x_i=1}, \qquad \sigma_i=\alpha_i+\beta_i.$$

Split $f=\sum_{\mathbf{x},y}\lambda_y\,\theta_{y|\mathbf{x}}\prod_i\lambda_{x_i}\theta_{x_i}$ according to whether $\mathbf{x}$ is all-false — on which block the inner product is $\prod_i\alpha_i$, while over all $\mathbf{x}$ the inner products sum to $\prod_i\sigma_i$, since $\sum_{x_i}\lambda_{x_i}\theta_{x_i}=\sigma_i$ and the sum factors — and apply the telescoping identity of Exercise 13.2 (with $\alpha_j,\beta_j,\sigma_j$ for $\lambda_{x_j=0},\lambda_{x_j=1},A_j$; the proof there uses only $\sigma_i-\alpha_i=\beta_i$):

$$\sum_{\substack{\mathbf{x}\\ x_i=1 \text{ for some } i}}\prod_{i=1}^n\lambda_{x_i}\theta_{x_i} \;=\; \prod_{i=1}^n\sigma_i-\prod_{i=1}^n\alpha_i \;=\; \sum_{i=1}^n\left(\prod_{j<i}\alpha_j\right)\beta_i\left(\prod_{j>i}\sigma_j\right).$$

Since $\theta_{y|\mathbf{x}}$ is constant on each block, this gives the factorization

$$f = \big(\epsilon_0\lambda_{y=0} + (1-\epsilon_0)\lambda_{y=1}\big)\prod_{i=1}^{n}\alpha_i + \big((1-\epsilon_1)\lambda_{y=0} + \epsilon_1\lambda_{y=1}\big)\sum_{i=1}^{n}\left(\prod_{j<i}\alpha_j\right)\beta_i\left(\prod_{j>i}\sigma_j\right),$$

realized term by term with leaves the $2n+2$ indicators, the $2n$ constants θ_{x_i} and $\epsilon_0, 1-\epsilon_0, \epsilon_1, 1-\epsilon_1$ (parameters substituted by value as in Figure 13.1(b)), and, exactly as in Exercise 13.1:

- $\alpha_i = \star(\lambda_{x_i=0}, \theta_{x_i=0})$, $\beta_i = \star(\lambda_{x_i=1}, \theta_{x_i=1})$: $2n$ multiplication nodes; $\sigma_i = +(\alpha_i, \beta_i)$: n addition nodes;
- the prefix chain $P_i = \star(P_{i-1}, \alpha_i) = \prod_{j\le i}\alpha_j$ with $P_0 = 1$: $n-1$ multiplication nodes; the suffix chain $S_i = \star(\sigma_i, S_{i+1}) = \prod_{j\ge i}\sigma_j$ with $S_{n+1} = 1$: $n-2$ multiplication nodes;
- the term nodes $T_i = \star(P_{i-1}, \beta_i, S_{i+1})$: at most $2n-2$ multiplication nodes; the accumulation $\Sigma = +(T_1, +(T_2, \ldots))$: $n-1$ addition nodes;
- the two Y-mixtures $M_0 = +(\star(\epsilon_0, \lambda_{y=0}), \star(1-\epsilon_0, \lambda_{y=1}))$ and $M_1 = +(\star(1-\epsilon_1, \lambda_{y=0}), \star(\epsilon_1, \lambda_{y=1}))$: 4 multiplication and 2 addition nodes;
- the root $f = +(\star(M_0, P_n), \star(M_1, \Sigma))$: 2 multiplication and 1 addition node.

The total is at most $6n+1$ multiplication nodes, $2n+2$ addition nodes and $4n+6$ leaves, hence $O(n)$ nodes and, all internal nodes being binary, $O(n)$ edges; the priors are unconstrained, entering only as the leaf constants inside α_i and β_i.

Problem 13.4. Consider a Bayesian network consisting of binary nodes $X_1, \ldots, X_n, Y$ and edges $X_i \to Y$ for $i = 1, \ldots, n$ (the structure of Figure 13.2: n roots X_i and their common child Y, whose treewidth is n). Assume that Y is true if and only if an odd number of its parents X_i are true, i.e. Y is the parity (exclusive-or) of its parents. Provide an arithmetic circuit for this network of size $O(n)$ while making no assumptions about the distributions over nodes X_i.

Solution. Parity admits no context-specific independence at all, yet the polynomial still has a linear circuit, because the parity of a prefix of the parents is a two-valued summary of that prefix and so obeys a two-line recurrence. With the values of X_i written $0, 1$, arbitrary priors $\theta_{x_i=0}, \theta_{x_i=1}$, and

$$\alpha_i = \lambda_{x_i=0}\,\theta_{x_i=0}, \qquad \beta_i = \lambda_{x_i=1}\,\theta_{x_i=1},$$

the deterministic CPT of Y — $\theta_{y=1|\mathbf{x}} = 1, \theta_{y=0|\mathbf{x}} = 0$ when $\bigoplus_i x_i = 1$, the reverse when $\bigoplus_i x_i = 0$ — kills every term of $f = \sum_{\mathbf{x},y}\lambda_y\theta_{y|\mathbf{x}}\prod_i \lambda_{x_i}\theta_{x_i}$ whose parent instantiation has the wrong parity, leaving $f = \lambda_{y=0}E_n + \lambda_{y=1}O_n$ for the even- and odd-parity partial sums

$$E_k = \sum_{\substack{x_1,\ldots,x_k \\ \bigoplus_{i\le k} x_i = 0}} \prod_{i\le k}\lambda_{x_i}\theta_{x_i}, \qquad O_k = \sum_{\substack{x_1,\ldots,x_k \\ \bigoplus_{i\le k} x_i = 1}} \prod_{i\le k}\lambda_{x_i}\theta_{x_i}.$$

Splitting on x_k — a prefix is even iff $x_k = 0$ with an even $(k-1)$-prefix or $x_k = 1$ with an odd one — gives, the empty instantiation having even parity,

$$\begin{aligned} E_0 &= 1, & O_0 &= 0, \\ E_k &= E_{k-1}\alpha_k + O_{k-1}\beta_k, & O_k &= O_{k-1}\alpha_k + E_{k-1}\beta_k \end{aligned}$$

for $k = 1, \ldots, n$. As in Figure 13.1(b) parameters are substituted by value, so the circuit takes the $2n+2$ indicators and the $2n$ constants θ_{x_i} as leaves, then the $2n$ products $\alpha_k = \star(\lambda_{x_k=0}, \theta_{x_k=0})$, $\beta_k = \star(\lambda_{x_k=1}, \theta_{x_k=1})$, then per stage k the four products $\star(E_{k-1}, \alpha_k)$, $\star(O_{k-1}, \beta_k)$, $\star(O_{k-1}, \alpha_k)$, $\star(E_{k-1}, \beta_k)$ and two addition nodes for E_k, O_k — only $n-1$ stages costing anything, stage 1 being degenerate with $E_1 = \alpha_1$, $O_1 = \beta_1$ — and the root $f = +(\star(\lambda_{y=0}, E_n), \star(\lambda_{y=1}, O_n))$. In all $2n + 4(n-1) + 2 = 6n-2$ multiplication nodes, $2(n-1)+1 = 2n-1$ addition nodes and $4n+2$ leaves, all internal nodes binary, so the size of Definition 12.2 is $O(n)$; nothing was assumed about the priors, which enter only as leaf constants.

Problem 13.5. Figure 13.20 depicts a Bayesian network with three nodes A, B, C and edges $A \to C$ and $B \to C$ (so A and B are roots and C has parents A, B). Variables A and B are binary with values a_1, a_2 and b_1, b_2; variable C has three values c_1, c_2, c_3. The CPT $\Theta_{C|AB}$ is

row	A	B	C	$\theta_{C\|AB}$
1	a_1	b_1	c_1	0
2	a_1	b_1	c_2	.5
3	a_1	b_1	c_3	.5
4	a_1	b_2	c_1	.2
5	a_1	b_2	c_2	.3
6	a_1	b_2	c_3	.5
7	a_2	b_1	c_1	0
8	a_2	b_1	c_2	0
9	a_2	b_1	c_3	1
10	a_2	b_2	c_1	.2
11	a_2	b_2	c_2	.3
12	a_2	b_2	c_3	.5

Construct two CNF encodings for this CPT, one that ignores local structure (Section 13.3.1) and another that accounts for local structure (Section 13.3.2).

Solution. Write I_x for the indicator variable of value x and $P_{x|\mathbf{u}}$ for the parameter variable of $\theta_{x|\mathbf{u}}$, with weights 1 on $I_x, \neg I_x, \neg P_{x|\mathbf{u}}$ and $\theta_{x|\mathbf{u}}$ on $P_{x|\mathbf{u}}$, so that weighted model counting gives probabilities of evidence (Section 11.6.1).

(1) Structure-based encoding (Section 13.3.1). Indicator clauses, the exactly-one constraint on C taking one positive clause and three pairwise exclusions:

$$\begin{aligned} &I_{a_1} \vee I_{a_2}, && \neg I_{a_1} \vee \neg I_{a_2}, \\ &I_{b_1} \vee I_{b_2}, && \neg I_{b_1} \vee \neg I_{b_2}, \\ &I_{c_1} \vee I_{c_2} \vee I_{c_3}, && \neg I_{c_1} \vee \neg I_{c_2}, \quad \neg I_{c_1} \vee \neg I_{c_3}, \quad \neg I_{c_2} \vee \neg I_{c_3}. \end{aligned}$$

Parameter clauses: one sentence per CPT row, twelve in all,

$$I_{a_i} \wedge I_{b_j} \wedge I_{c_k} \iff P_{c_k|a_i b_j}, \qquad i, j \in \{1, 2\},\ k \in \{1, 2, 3\},$$

each of which unfolds (as in footnote 2 of Section 13.3.1) into one IP clause

$$\neg I_{a_i} \vee \neg I_{b_j} \vee \neg I_{c_k} \vee P_{c_k|a_i b_j}$$

and three PI clauses

$$\neg P_{c_k|a_i b_j} \vee I_{a_i}, \qquad \neg P_{c_k|a_i b_j} \vee I_{b_j}, \qquad \neg P_{c_k|a_i b_j} \vee I_{c_k}.$$

In all 7 indicator and 12 parameter variables and $8 + 12 + 36 = 56$ clauses, the numbers entering only as literal weights.

(2) Encoding with local structure (Section 13.3.2). Three refinements in turn.

Zero parameters. Rows $1, 7, 8$ are zero; each drops its parameter variable and its clauses in favour of the single clause $\neg I_{u_1} \vee \ldots \vee \neg I_{u_n} \vee \neg I_x$:

$$\neg I_{a_1} \vee \neg I_{b_1} \vee \neg I_{c_1}, \qquad \neg I_{a_2} \vee \neg I_{b_1} \vee \neg I_{c_1}, \qquad \neg I_{a_2} \vee \neg I_{b_1} \vee \neg I_{c_2}.$$

Resolving the first two against $I_{a_1} \vee I_{a_2}$ replaces them by the equivalent shorter $\neg I_{b_1} \vee \neg I_{c_1}$.

One parameters. Row 9 has $\theta_{c_3|a_2 b_1} = 1$, so its positive literal has weight 1 and can neither reweight nor exclude a model: $P_{c_3|a_2 b_1}$ and its four clauses are dropped, losing nothing logically either, since given a_2, b_1 the zero clauses force $\neg I_{c_1}, \neg I_{c_2}$ and then $I_{c_1} \vee I_{c_2} \vee I_{c_3}$ forces I_{c_3}.

Equal parameters. The eight remaining rows take three distinct values, so introduce one variable per group:

- η_1, weight .5, for rows $2, 3, 6, 12$;
- η_2, weight .2, for rows $4, 10$;
- η_3, weight .3, for rows $5, 11$.

The defining sentences are $(I_{f_1} \vee \ldots \vee I_{f_m}) \iff \eta$, which here read

$$\begin{aligned}(I_{a_1} \wedge I_{b_1} \wedge I_{c_2}) \vee (I_{a_1} \wedge I_{b_1} \wedge I_{c_3}) \vee (I_{a_1} \wedge I_{b_2} \wedge I_{c_3}) \vee (I_{a_2} \wedge I_{b_2} \wedge I_{c_3}) &\iff \eta_1,\\ (I_{a_1} \wedge I_{b_2} \wedge I_{c_1}) \vee (I_{a_2} \wedge I_{b_2} \wedge I_{c_1}) &\iff \eta_2,\\ (I_{a_1} \wedge I_{b_2} \wedge I_{c_2}) \vee (I_{a_2} \wedge I_{b_2} \wedge I_{c_2}) &\iff \eta_3.\end{aligned}$$

The indicator clauses simplify these. In η_2, η_3 the two disjuncts differ only in A, and $I_{a_1} \vee I_{a_2}$ is a clause, so

$$\eta_2 \iff I_{b_2} \wedge I_{c_1}, \qquad \eta_3 \iff I_{b_2} \wedge I_{c_2}.$$

In η_1 the last two disjuncts collapse the same way to $I_{b_2} \wedge I_{c_3}$, while the first two collapse to $I_{a_1} \wedge I_{b_1} \wedge (I_{c_2} \vee I_{c_3})$, and $I_{c_2} \vee I_{c_3}$ is equivalent to $\neg I_{c_1}$ under the exactly-one constraint on C. Hence

$$\eta_1 \iff (I_{a_1} \wedge I_{b_1} \wedge \neg I_{c_1}) \vee (I_{b_2} \wedge I_{c_3}).$$

Converting to CNF (Section 2.7.3), for η_2, η_3 immediately:

$$\begin{aligned}&\neg\eta_2 \vee I_{b_2}, \quad \neg\eta_2 \vee I_{c_1}, \quad \neg I_{b_2} \vee \neg I_{c_1} \vee \eta_2,\\ &\neg\eta_3 \vee I_{b_2}, \quad \neg\eta_3 \vee I_{c_2}, \quad \neg I_{b_2} \vee \neg I_{c_2} \vee \eta_3.\end{aligned}$$

For η_1, the direction "disjunct implies η_1" gives the two clauses

$$\neg I_{a_1} \vee \neg I_{b_1} \vee I_{c_1} \vee \eta_1, \qquad \neg I_{b_2} \vee \neg I_{c_3} \vee \eta_1,$$

and the direction $\eta_1 \Rightarrow (u \vee v)$, with $u = I_{a_1} \wedge I_{b_1} \wedge \neg I_{c_1}$ and $v = I_{b_2} \wedge I_{c_3}$, is put in CNF by distributing the six literal pairs:

$$\begin{aligned}&\neg\eta_1 \vee I_{a_1} \vee I_{b_2}, \qquad \neg\eta_1 \vee I_{a_1} \vee I_{c_3},\\ &\neg\eta_1 \vee I_{b_1} \vee I_{c_3}, \qquad \neg\eta_1 \vee \neg I_{c_1} \vee I_{b_2},\\ &\neg\eta_1 \vee \neg I_{c_1} \vee I_{c_3}.\end{aligned}$$

(The sixth product, $\neg\eta_1 \vee I_{b_1} \vee I_{b_2}$, is subsumed by the indicator clause $I_{b_1} \vee I_{b_2}$ and is dropped.)
Putting it together, the local-structure encoding consists of the 8 indicator clauses, the determinism clauses $\neg I_{b_1} \vee \neg I_{c_1}$ and $\neg I_{a_2} \vee \neg I_{b_1} \vee \neg I_{c_2}$, and the 13 clauses above, using 7 indicator variables and only 3 parameter variables η_1, η_2, η_3 of weights $.5, .2, .3$: 23 clauses and 10 variables, against 56 clauses and 19 variables for the structure-based encoding.
Each instantiation $a_i b_j c_k$ fixes the indicators, hence the η's, hence at most one model, of weight the product of the weights of the η's set true; walking the rows recovers the twelve CPT entries (Check!).

Problem 13.6. Consider again the Bayesian network of Figure 13.20: nodes A, B, C with edges $A \to C$ and $B \to C$, where A is binary with values a_1, a_2, B is binary with values b_1, b_2, and C has three values c_1, c_2, c_3, with CPT

row	A	B	C	$\theta_{C\|AB}$
1	a_1	b_1	c_1	0
2	a_1	b_1	c_2	.5
3	a_1	b_1	c_3	.5
4	a_1	b_2	c_1	.2
5	a_1	b_2	c_2	.3
6	a_1	b_2	c_3	.5
7	a_2	b_1	c_1	0
8	a_2	b_1	c_2	0
9	a_2	b_1	c_3	1
10	a_2	b_2	c_1	.2
11	a_2	b_2	c_2	.3
12	a_2	b_2	c_3	.5

Construct an arithmetic circuit for this network while assuming that the instantiation a_2, b_1 is always part of the evidence (see Section 13.3.3). Assume that variables A and B have uniform CPTs, that is, $\theta_{a_1} = \theta_{a_2} = \theta_{b_1} = \theta_{b_2} = .5$.

Solution. Building the evidence in collapses the circuit to the single multiplication node $.25\,\lambda_{c_3}$. By the second option of Section 13.3.3 — encode in CNF, condition, simplify, compile to a decomposable, deterministic, smooth NNF circuit, extract by Section 12.4.3:
(1) CNF with local structure. The $\Theta_{C|AB}$ clauses are those of Exercise 13.5; the two uniform root CPTs are equal parameters, so become P_A and P_B of weight .5. In full:

- indicator clauses: $I_{a_1} \vee I_{a_2}$, $\neg I_{a_1} \vee \neg I_{a_2}$, $I_{b_1} \vee I_{b_2}$, $\neg I_{b_1} \vee \neg I_{b_2}$, $I_{c_1} \vee I_{c_2} \vee I_{c_3}$ and the three pairwise exclusions on C;
- determinism: $\neg I_{b_1} \vee \neg I_{c_1}$ and $\neg I_{a_2} \vee \neg I_{b_1} \vee \neg I_{c_2}$;
- $\eta_1 \iff (I_{a_1} \wedge I_{b_1} \wedge \neg I_{c_1}) \vee (I_{b_2} \wedge I_{c_3})$, weight .5;
- $\eta_2 \iff I_{b_2} \wedge I_{c_1}$, weight .2;
- $\eta_3 \iff I_{b_2} \wedge I_{c_2}$, weight .3;
- $P_A \iff I_{a_1} \vee I_{a_2}$, weight .5; $P_B \iff I_{b_1} \vee I_{b_2}$, weight .5.

(2) Condition. Replacing I_{a_2}, I_{b_1} by $\top$, unit propagation forces $\neg I_{a_1}$ and $\neg I_{b_2}$ from the exclusion clauses, then $\neg I_{c_1}$ and $\neg I_{c_2}$ from the two determinism clauses, then I_{c_3} from $I_{c_1} \vee I_{c_2} \vee I_{c_3}$, then $\neg\eta_1, \neg\eta_2, \neg\eta_3$ (every disjunct needs I_{a_1} or I_{b_2}), and P_A, P_B. Every variable is determined, so the conditioned CNF has the single model

$$\neg I_{a_1},\ \neg I_{b_2},\ \neg I_{c_1},\ \neg I_{c_2},\ I_{c_3},\ \neg\eta_1,\ \neg\eta_2,\ \neg\eta_3,\ P_A,\ P_B.$$

(3) Compile and extract. A CNF with a single model is the conjunction of that model's literals,

$$\Gamma = \neg I_{a_1} \wedge \neg I_{b_2} \wedge \neg I_{c_1} \wedge \neg I_{c_2} \wedge I_{c_3} \wedge \neg\eta_1 \wedge \neg\eta_2 \wedge \neg\eta_3 \wedge P_A \wedge P_B,$$

a single and-node over distinct variables — decomposable, deterministic for want of an or-node, and smooth. The substitutions $\wedge \mapsto \star$, negative literal $\mapsto 1$, $I_x \mapsto \lambda_x$, $P \mapsto$ its weight then give

$$\mathrm{AC} \;=\; \star(1,1,1,1,\lambda_{c_3},1,1,1,.5,.5) \;=\; \star(.5,\ .5,\ \lambda_{c_3}),$$

that is, after folding the constants, one multiplication node with leaves .25 and λ_{c_3}, representing $f^{a_2,b_1} = .25\,\lambda_{c_3}$ — which is what setting $\lambda_{a_2} = \lambda_{b_1} = 1$, $\lambda_{a_1} = \lambda_{b_2} = 0$ in $f = \sum_{a,b,c} \lambda_a\lambda_b\lambda_c\,\theta_a\theta_b\theta_{c|ab}$ leaves directly, $\theta_{a_2}\theta_{b_1}\lambda_{c_3} \cdot 1 = .25\,\lambda_{c_3}$. The price is that the A- and B-indicators no longer occur, so the circuit answers only queries with a_2, b_1 in the evidence.

Problem 13.7. Figure 13.21 gives the conditional probability table $\Theta_{D|A,B,C}$ of a binary variable D with binary parents A, B and C:

A	B	C	D	$\Pr(D \mid A, B, C)$
true	true	true	true	.9
true	true	true	false	.1
true	true	false	true	.9
true	true	false	false	.1
true	false	true	true	.9
true	false	true	false	.1
true	false	false	true	.2
true	false	false	false	.8
false	true	true	true	.5
false	true	true	false	.5
false	true	false	true	.5
false	true	false	false	.5
false	false	true	true	0
false	false	true	false	1
false	false	false	true	0
false	false	false	false	1

Encode this CPT as a CNF while ignoring local structure (Section 13.3.1). Show how the encoding would change if we account for determinism (Section 13.3.2). Show how this last encoding would change if we account for equal parameters (Section 13.3.2).

Solution. Write $a, \bar{a}$ for the values of A and similarly for B, C, D; indicator literals and negative parameter literals have weight 1, a positive parameter literal the weight of its parameter.
(1) Ignoring local structure. Indicator clauses, two per variable:

$$I_a \vee I_{\bar{a}}, \quad \neg I_a \vee \neg I_{\bar{a}}, \quad I_b \vee I_{\bar{b}}, \quad \neg I_b \vee \neg I_{\bar{b}},$$
$$I_c \vee I_{\bar{c}}, \quad \neg I_c \vee \neg I_{\bar{c}}, \quad I_d \vee I_{\bar{d}}, \quad \neg I_d \vee \neg I_{\bar{d}}.$$

Parameter clauses: one sentence per CPT row, sixteen in all,

$$I_{a'} \wedge I_{b'} \wedge I_{c'} \wedge I_{d'} \iff P_{d'|a'b'c'},$$

where a' ranges over $a, \bar{a}$ and so on. Each sentence unfolds into one IP clause

$$\neg I_{a'} \vee \neg I_{b'} \vee \neg I_{c'} \vee \neg I_{d'} \vee P_{d'|a'b'c'}$$

and four PI clauses $\neg P_{d'|a'b'c'} \vee I_{a'}$, $\neg P_{d'|a'b'c'} \vee I_{b'}$, $\neg P_{d'|a'b'c'} \vee I_{c'}$, $\neg P_{d'|a'b'c'} \vee I_{d'}$. The weight of $P_{d'|a'b'c'}$ is the number in the corresponding row.
In all 8 indicator and 16 parameter variables and $8 + 16 + 64 = 88$ clauses, depending only on the structure — the numbers appear only as weights.
(2) Accounting for determinism. The zero parameters $\theta_{d|\bar{a}\bar{b}c}$ and $\theta_{d|\bar{a}\bar{b}\bar{c}}$ lose their variables and ten clauses in favour of

$$\neg I_{\bar{a}} \vee \neg I_{\bar{b}} \vee \neg I_c \vee \neg I_d, \qquad \neg I_{\bar{a}} \vee \neg I_{\bar{b}} \vee \neg I_{\bar{c}} \vee \neg I_d.$$

which resolve on C and, since $I_c \vee I_{\bar{c}}$ is a clause, are jointly equivalent to the shorter $\neg I_{\bar{a}} \vee \neg I_{\bar{b}} \vee \neg I_d$ — if A and B are both false then so is D, whatever C is. The one parameters $\theta_{\bar{d}|\bar{a}\bar{b}c}$ and $\theta_{\bar{d}|\bar{a}\bar{b}\bar{c}}$ have their variables and ten clauses dropped outright, losing nothing, since given $\bar{a}, \bar{b}$ that clause forces $\neg I_d$ and then $I_d \vee I_{\bar{d}}$ forces $I_{\bar{d}}$. The encoding is now 8 indicator and 12 parameter variables with $8+1+12 \cdot 5 = 69$ clauses: four CPT rows replaced by one clause.
(3) Also accounting for equal parameters. The twelve surviving parameters take five distinct values:

value	parent-and-value instantiations
.9	$abcd$, $ab\bar{c}d$, $a\bar{b}cd$
.1	$abc\bar{d}$, $ab\bar{c}\bar{d}$, $a\bar{b}c\bar{d}$
.2	$a\bar{b}\bar{c}d$
.8	$a\bar{b}\bar{c}\bar{d}$
.5	$\bar{a}bcd$, $\bar{a}bc\bar{d}$, $\bar{a}b\bar{c}d$, $\bar{a}b\bar{c}\bar{d}$

so five variables $\eta_{.9}, \eta_{.1}, \eta_{.2}, \eta_{.8}, \eta_{.5}$ of those weights replace the twelve P variables, with defining sentences $(I_{f_1} \vee \ldots \vee I_{f_m}) \iff \eta$ that the indicator clauses simplify:

$$\begin{aligned}\eta_{.9} &\iff (I_a \wedge I_b \wedge I_c \wedge I_d) \vee (I_a \wedge I_b \wedge I_{\bar{c}} \wedge I_d) \vee (I_a \wedge I_{\bar{b}} \wedge I_c \wedge I_d) \\ &\iff I_a \wedge I_d \wedge \big(I_b \wedge (I_c \vee I_{\bar{c}}) \;\vee\; I_{\bar{b}} \wedge I_c\big) \\ &\iff I_a \wedge I_d \wedge (I_b \vee I_c),\end{aligned}$$

using $I_c \vee I_{\bar{c}}$ and absorbing $I_{\bar{b}} \wedge I_c$ into I_c; identically,

$$\eta_{.1} \iff I_a \wedge I_{\bar{d}} \wedge (I_b \vee I_c).$$

The .5 group holds all four combinations of C, D under $\bar{a}, b$, so

$$\eta_{.5} \iff I_{\bar{a}} \wedge I_b \wedge (I_c \vee I_{\bar{c}}) \wedge (I_d \vee I_{\bar{d}}) \iff I_{\bar{a}} \wedge I_b.$$

The two singleton groups keep their full conjunctions,

$$\eta_{.2} \iff I_a \wedge I_{\bar{b}} \wedge I_{\bar{c}} \wedge I_d, \qquad \eta_{.8} \iff I_a \wedge I_{\bar{b}} \wedge I_{\bar{c}} \wedge I_{\bar{d}}.$$

In CNF (Section 2.7.3):

$$\begin{aligned}\eta_{.5}: &\quad \neg\eta_{.5} \vee I_{\bar{a}}, \quad \neg\eta_{.5} \vee I_b, \quad \neg I_{\bar{a}} \vee \neg I_b \vee \eta_{.5}; \\ \eta_{.2}: &\quad \neg\eta_{.2} \vee I_a, \quad \neg\eta_{.2} \vee I_{\bar{b}}, \quad \neg\eta_{.2} \vee I_{\bar{c}}, \quad \neg\eta_{.2} \vee I_d, \\ &\quad \neg I_a \vee \neg I_{\bar{b}} \vee \neg I_{\bar{c}} \vee \neg I_d \vee \eta_{.2}; \\ \eta_{.8}: &\quad \text{the same four unit-implication clauses with } I_{\bar{d}} \text{ in place of } I_d, \\ &\quad \neg I_a \vee \neg I_{\bar{b}} \vee \neg I_{\bar{c}} \vee \neg I_{\bar{d}} \vee \eta_{.8}; \\ \eta_{.9}: &\quad \neg\eta_{.9} \vee I_a, \quad \neg\eta_{.9} \vee I_d, \quad \neg\eta_{.9} \vee I_b \vee I_c, \\ &\quad \neg I_a \vee \neg I_d \vee \neg I_b \vee \eta_{.9}, \quad \neg I_a \vee \neg I_d \vee \neg I_c \vee \eta_{.9}; \\ \eta_{.1}: &\quad \text{the same five clauses with } I_{\bar{d}} \text{ in place of } I_d.\end{aligned}$$

The final encoding is 8 indicator clauses, the determinism clause $\neg I_{\bar{a}} \vee \neg I_{\bar{b}} \vee \neg I_d$ and $3{+}5{+}5{+}5{+}5 = 23$ parameter clauses: 32 clauses over 13 variables, against 88 over 24 in (1) and 69 over 20 in (2).
At most one η is ever true — $\eta_{.5}$ demands $I_{\bar{a}}$ and the others I_a; $\eta_{.9}, \eta_{.2}$ demand I_d and $\eta_{.1}, \eta_{.8}$ demand $I_{\bar{d}}$; and within $\{\eta_{.9}, \eta_{.2}\}$ and $\{\eta_{.1}, \eta_{.8}\}$ the first demands $I_b \vee I_c$ and the second $I_{\bar{b}} \wedge I_{\bar{c}}$ — so each instantiation of A, B, C, D has model weight the weight of its unique true η, or 1 if there is none, which walking the blocks reproduces row by row (Check!).

12.2 Exercises 13.8–13.14

Problem 13.8. Construct an ADD for the CPT in Figure 13.21 using the variable order A, B, C, D. Figure 13.21 is the conditional probability table $\Pr(D \mid A, B, C)$ over binary variables:

A	B	C	D	$\Pr(D \mid A, B, C)$
true	true	true	true	.9
true	true	true	false	.1
true	true	false	true	.9
true	true	false	false	.1
true	false	true	true	.9
true	false	true	false	.1
true	false	false	true	.2
true	false	false	false	.8
false	true	true	true	.5
false	true	true	false	.5
false	true	false	true	.5
false	true	false	false	.5
false	false	true	true	0
false	false	true	false	1
false	false	false	true	0
false	false	false	false	1

Recall the ADD conventions of Section 13.5.1: every nonleaf node is labeled with a binary variable and has a *high* child (solid edge, taken when the variable is true) and a *low* child (dotted edge, taken when the variable is false); leaves (sinks) are labeled with real numbers; variables must occur in the fixed order along every root-to-leaf path; and the ADD is to be reduced, i.e. it contains no node whose high and low children are identical and no two distinct nodes with the same label and the same pair of children.

Solution. Read the table as a factor $f(A, B, C, D)$ and group its rows by the context A, B, C. Each context fixes a pair of numbers (f at $D = \text{false}$, f at $D = \text{true}$):

A	B	C	($D = \text{false}$, $D = \text{true}$)
true	true	true	(.1, .9)
true	true	false	(.1, .9)
true	false	true	(.1, .9)
true	false	false	(.8, .2)
false	true	true	(.5, .5)
false	true	false	(.5, .5)
false	false	true	(1, 0)
false	false	false	(1, 0)

Four distinct pairs occur, $(.1, .9), (.8, .2), (.5, .5), (1, 0)$, and $(.5, .5)$ has equal components so its node is redundant and collapses to the sink .5. This leaves three D-nodes:

$$\delta_1 = (\text{low } .1,\ \text{high } .9), \quad \delta_2 = (\text{low } .8,\ \text{high } .2), \quad \delta_3 = (\text{low } 1,\ \text{high } 0).$$

At the C-level the four contexts A, B give: (i) A, B true, both C values reach δ_1, so the node collapses to δ_1; (ii) A true, B false, the two values reach δ_1 and δ_2, so a genuine node $\gamma = (\text{low } \delta_2, \text{high } \delta_1)$ survives; (iii) A false, B true, both reach the sink .5; (iv) both false, both reach δ_3. One level up, the B-nodes are

$$\beta_1 = (\text{low } \gamma,\ \text{high } \delta_1), \qquad \beta_2 = (\text{low } \delta_3,\ \text{high } .5),$$

both genuine and distinct, and the root is $\alpha = (\text{low } \beta_2, \text{high } \beta_1)$. The reduced ADD, node by node (a dash marks a nonleaf in the value column, a sink in the child columns):

node	label	low child (dotted)	high child (solid)	value
α	A	β_2	β_1	–
β_1	B	γ	δ_1	–
β_2	B	δ_3	$s_{.5}$	–
γ	C	δ_2	δ_1	–
δ_1	D	$s_{.1}$	$s_{.9}$	–
δ_2	D	$s_{.8}$	$s_{.2}$	–
δ_3	D	s_1	s_0	–
$s_{.1}$	–	–	–	.1
$s_{.9}$	–	–	–	.9
$s_{.8}$	–	–	–	.8
$s_{.2}$	–	–	–	.2
$s_{.5}$	–	–	–	.5
s_1	–	–	–	1
s_0	–	–	–	0

Two edges skip a variable, $\beta_1 \to \delta_1$ skipping C and $\beta_2 \to s_{.5}$ skipping C and D, legal since a path need not mention every variable of the order; they record the context-specific independences of D from C given A, B true and of D from B, C given A false, B true. The ADD has 7 nonleaf nodes and 7 sinks against the table's 16 rows, and is reduced (Check!), hence by the canonicity of Section 13.5.1 the *unique* ADD for this factor under A, B, C, D.

Problem 13.9. Consider a factor f over binary variables $X_1, \ldots, X_n$, where $f(x_1, \ldots, x_n) = 1$ if exactly one value in $x_1, \ldots, x_n$ is true and $f(x_1, \ldots, x_n) = 0$ otherwise. Construct an ADD representation of this factor with size $O(n)$.

Solution. Under the order $X_1 \prec \cdots \prec X_n$, what the remaining variables must do depends on the prefix only through a two-valued summary — the number of true values so far, capped at two — so two nodes per level suffice. For $i \in \{1, \ldots, n+1\}$ and $c \in \{0, 1\}$ put

$$g_i^c(x_i, \ldots, x_n) \;=\; \begin{cases} 1, & \text{if exactly } 1-c \text{ of } x_i, \ldots, x_n \text{ are true,} \\ 0, & \text{otherwise.} \end{cases}$$

with c the number of true values among $X_1, \ldots, X_{i-1}$; $c \geq 2$ makes the residual identically 0, so only $c \in \{0, 1\}$ is needed, and $f = g_1^0$. The Shannon expansions are

$$g_i^0 = \overline{X_i}\, g_{i+1}^0 + X_i\, g_{i+1}^1,$$
$$g_i^1 = \overline{X_i}\, g_{i+1}^1 + X_i\, 0,$$

for $1 \leq i \leq n$, with $g_{n+1}^0 = 0$ and $g_{n+1}^1 = 1$. Writing A_i, B_i for the nodes labelled X_i representing g_i^0, g_i^1 and s_0, s_1 for the sinks, each expansion becomes one node:

node	label	low child (dotted, X_i = false)	high child (solid, X_i = true)
A_i, $1 \leq i \leq n-1$	X_i	A_{i+1}	B_{i+1}
B_i, $2 \leq i \leq n-1$	X_i	B_{i+1}	s_0
A_n	X_n	s_0	s_1
B_n	X_n	s_1	s_0

with root A_1; B_1 is never created, since nothing precedes X_1. A path runs down the A-chain while every variable seen is false, the first true variable moves it onto the B-chain, and any further true variable drops it to s_0; so a path reaches s_1 exactly when it takes precisely one solid edge, and the ADD represents f.

The nodes are $A_1, \ldots, A_n$, $B_2, \ldots, B_n$ and the two sinks: $2n+1$ nodes and $4n-2$ edges, against the 2^n rows of the table. It is reduced — no node has identical children (the only candidates, A_i with $i \leq n-1$, have the distinct A_{i+1}, B_{i+1}), and the same-label pair A_i, B_i always differs, since $A_n = (s_0, s_1)$ against $B_n = (s_1, s_0)$ while for $i \leq n-1$ the high child of A_i is a nonleaf and that of B_i the sink s_0.

Problem 13.10. Reduce the ADD in Figure 13.22 (see also Figure 13.14).
Figure 13.22 is an unreduced ADD – in fact a complete decision tree – over the binary variables X, Y, Z in the order X, Y, Z. Its root is labeled X; both of the root's children are labeled Y; all four grandchildren are labeled Z; and each of the four Z-nodes points to two sinks of its own, eight sinks in all. Reading the eight sinks left to right (a left/dotted edge means the variable is false, a right/solid edge means it is true), the tree is

X	Y	Z	f
false	false	false	.1
false	false	true	.9
false	true	false	.1
false	true	true	.9
true	false	false	.9
true	false	true	.1
true	true	false	.5
true	true	true	.5

Figure 13.14 illustrates the reduction procedure: an ADD node is *redundant* if its high and low children are the same node, in which case the node is deleted and its parents are redirected to that common child; or if some other node in the ADD carries the same label and the same high and low children, in which case the node is deleted and its parents are redirected to that duplicate. A reduced ADD contains no redundant nodes.

Solution. Reduce bottom-up, since a deletion can only expose further merges above it. The eight sinks carry three distinct labels, so they merge to $s_{.1}, s_{.9}, s_{.5}$, and the four Z-nodes then have child pairs (low, high):

context	Z-node child pair
$X = \text{false}, Y = \text{false}$	$(s_{.1},\ s_{.9})$
$X = \text{false}, Y = \text{true}$	$(s_{.1},\ s_{.9})$
$X = \text{true}, Y = \text{false}$	$(s_{.9},\ s_{.1})$
$X = \text{true}, Y = \text{true}$	$(s_{.5},\ s_{.5})$

The fourth is redundant by the first rule and is replaced by $s_{.5}$; the first two are duplicates by the second rule and merge into $\zeta_1 = (s_{.1}, s_{.9})$; the third survives as $\zeta_2 = (s_{.9}, s_{.1})$, no duplicate of ζ_1 since the child pair is *ordered*. At the Y-level, (i) under X false the node is (ζ_1, ζ_1), redundant, so the root's low edge goes straight to ζ_1; (ii) under X true it is $\eta = (\zeta_2, s_{.5})$, which survives, as does the root $\xi = (\zeta_1, \eta)$. The reduced ADD has four nonleaf nodes and three sinks, down from fifteen:

node	label	low child (dotted)	high child (solid)
ξ	X	ζ_1	η
η	Y	ζ_2	$s_{.5}$
ζ_1	Z	$s_{.1}$	$s_{.9}$
ζ_2	Z	$s_{.9}$	$s_{.1}$

with sinks $s_{.1} = .1$, $s_{.9} = .9$, $s_{.5} = .5$ — the ADD of Figure 13.15(a) for f_1. Tracing the eight instantiations:

X	Y	Z	path	value
false	false	false	$\xi \to \zeta_1 \to s_{.1}$	.1
false	false	true	$\xi \to \zeta_1 \to s_{.9}$	.9
false	true	false	$\xi \to \zeta_1 \to s_{.1}$	.1
false	true	true	$\xi \to \zeta_1 \to s_{.9}$	.9
true	false	false	$\xi \to \eta \to \zeta_2 \to s_{.9}$	.9
true	false	true	$\xi \to \eta \to \zeta_2 \to s_{.1}$	.1
true	true	false	$\xi \to \eta \to s_{.5}$	.5
true	true	true	$\xi \to \eta \to s_{.5}$	.5

which agree with the eight leaves of Figure 13.22; the result is reduced (Check!), hence by the canonicity of Section 13.5.1 the unique ADD for this factor under X, Y, Z.

Problem 13.11. Show how we can reduce an ADD with n nodes using an algorithm that takes $O(n \log n)$ time (see Figure 13.14).
Recall from Section 13.5.1 that a node of an ADD is *redundant* if its high and low children are the same node, or if another node in the ADD carries the same label and the same high and low children; reducing an ADD means repeatedly deleting redundant nodes, redirecting their parents to the surviving child or duplicate, until none remains. Figure 13.14 shows the successive removals on a small example.

Solution. Sweep the diagram once *bottom-up* in the variable order, giving every node a canonical integer identifier and detecting duplicates at each level by sorting rather than pairwise comparison; the naive scan-delete-rescan is $\Theta(n^2)$ because each deletion can expose redundancies higher up.
Setup. For the order $X_1 \prec \cdots \prec X_k$, bucket the n nodes of φ by label into levels, level $k+1$ the sinks and level i the nodes labelled X_i, in $O(n)$ by one traversal; since φ respects $\prec$, every child of a level-i node lies at a level $> i$. Write n_i for the size of level i, so $\sum_i n_i = n$. We build maps $\mathrm{id}(\cdot)$ to integers and $\mathrm{rep}(\cdot)$ to output nodes under the invariant that, for processed u, v, $\mathrm{id}(u) = \mathrm{id}(v)$ iff the sub-ADDs at u and v denote the same factor, and $\mathrm{rep}(u)$ is the unique output node with that identifier.

1. *Sinks.* Sort the n_{k+1} sinks by numeric label in $O(n_{k+1} \log n_{k+1})$ and scan, creating one output sink per distinct label and giving every input sink with that label the same fresh identifier and rep. Two sinks denote the same constant factor exactly when their labels agree.
2. *Levels* $i = k, \ldots, 1$. Every child of a level-i node lies later and is already processed. For v with children ℓ, h: if $\mathrm{id}(\ell) = \mathrm{id}(h)$ then v denotes the same factor independently of X_i, so set $\mathrm{id}(v) \leftarrow \mathrm{id}(\ell)$, $\mathrm{rep}(v) \leftarrow \mathrm{rep}(\ell)$ and delete v — the first rule, applied by identifier rather than pointer, which is what lets deletions cascade upward in one pass. Otherwise give v the signature $\sigma(v) = (\mathrm{id}(\ell), \mathrm{id}(h))$. Sort the level's survivors lexicographically by signature in $O(n_i \log n_i)$ and scan: each maximal run of equal signatures is a duplicate class, which gets one output node labelled X_i with children $\mathrm{rep}(\ell), \mathrm{rep}(h)$, a common fresh identifier, and that node as every member's rep — the second rule, applied to the whole level at once.
3. Return $\mathrm{rep}(\varphi)$.

Correctness. The invariant holds at the sinks by step 1. Assuming it above level i, two level-i nodes u, v denote the Shannon expansions

$$f_v \;=\; \overline{X_i}\, f_{\mathrm{low}(v)} \;+\; X_i\, f_{\mathrm{high}(v)},$$

so restricting to X_i false and true gives $f_u = f_v$ iff the corresponding children agree, which by the induction hypothesis is $\sigma(u) = \sigma(v)$; and a node deleted by the first test has $f_v = f_{\mathrm{low}(v)}$, so inheriting its child's identifier is correct too. Different levels never share an identifier, legitimately: a surviving level-i node denotes a factor that genuinely depends on X_i (its two restrictions differ), whereas a later-level node denotes one not mentioning X_i. At the root, then, the output denotes the input's factor, and it is reduced — no output node has two children of equal identifier and no two share a label and child pair — hence by canonicity the unique reduced ADD under $\prec$.
Complexity. Bucketing costs $O(n)$ and every node is touched a constant number of times outside the sorts, which cost

$$\sum_{i=1}^{k+1} O(n_i \log n_i) \;\le\; O\left(\log n \sum_{i=1}^{k+1} n_i\right) \;=\; O(n \log n),$$

using $n_i \le n$; everything else is $O(1)$ per node or edge, and the ADD has at most $2n$ edges. Total: $O(n \log n)$ time, $O(n)$ space. (Identifiers issued consecutively lie in $\{1, \ldots, n\}$, so counting sort on signatures gives an $O(n)$ worst-case variant and hashing an $O(n)$ expected one; the comparison version assumes nothing about identifiers or hashing.)

Problem 13.12. Consider the method described in Section 13.5.2 for converting a tabular representation of a factor into its ADD representation. Consider now a second method in which we

construct a decision tree to represent the factor (see Figure 13.22) and then reduce the ADD using the algorithm developed in Exercise 13.11. Implement and compare the efficiency of these methods. Recall the first method: for each row of the table one builds an ADD that maps the instantiation of that row to the row's value and every other instantiation to 0 – a single path of k nonleaf nodes hanging off a sink 0, as in Figure 13.17(a) and (b) – and then adds all these row ADDs together with the APPLY operation of Algorithm 37, as in Figure 13.17(c).

Solution. Method 2 dominates: $O(N \log N)$, or $O(N)$ with radix sorting, against Method 1's quadratic cost, where $N = 2^k$ is the number of rows for a factor over k binary variables.
Method 1 (row ADDs plus APPLY). Building the N row ADDs, each k nonleaf nodes and two sinks, costs $O(kN)$; the N accumulating APPLY calls dominate. With φ_j the accumulator after j rows, an APPLY of sizes a, b costs $O(ab)$ by Algorithm 37 and the second argument is a path of $O(k)$ nodes, so one call costs $O(k\,|\varphi_j|)$ — the cache is indexed by pairs (node of φ_j, node of the path), and the path has one node per level. Since $|\varphi_j|$ grows until it saturates at the final ADD's size,

$$\sum_{j=1}^{N} O(k\,|\varphi_j|) \;=\; O\Big(kN \max_j |\varphi_j|\Big) \;=\; O(kN^2),$$

quadratic in the table size, with a large constant besides: each APPLY allocates a fresh cache, rebuilds every accumulator node, and does a unique-table lookup per node.
Method 2 (decision tree plus reduce). The complete decision tree is one bottom-up pass — level $k+1$ is the N sinks read off the table, each higher level pairs adjacent entries below — giving $2N-1$ nodes in $O(N)$; reducing it by Exercise 13.11 costs $O(n \log n)$ with $n = 2N-1$,

$$O(N \log N) \;=\; O(2^k\, k),$$

dropping to $O(N)$ with the radix-sort variant, which is optimal since the table must be read; and the tree need never be materialized, since reducing each level as it is built keeps peak memory to the reduced levels alone.
Measurements. Both methods were implemented over a common ADD manager with a unique table and the cached APPLY of Algorithm 37, Method 2 using the hashing variant of Exercise 13.11. The instrumented cost is node-processing steps: for Method 1 the nodes built for the N row ADDs plus the distinct APPLY cache entries; for Method 2 the N sinks plus the $N-1$ constructions of the sweep. Values were drawn from $\{0, .1, .2, .5, .8, .9, 1\}$, and "dep" is the number of leading variables the factor genuinely depends on, so dep $= k$ has no local structure and dep $= k/2$ a great deal. Times are seconds on one core.

k	dep	rows N	Method 1 steps	Method 2 steps	Method 1 sec	Method 2 sec
6	6	64	2,596	127	0.006	0.000
6	3	64	1,223	127	0.002	0.000
8	8	256	28,550	511	0.044	0.000
8	4	256	6,832	511	0.007	0.000
10	10	1,024	315,741	2,047	0.498	0.001
10	5	1,024	37,761	2,047	0.039	0.000
12	12	4,096	4,147,677	8,191	3.750	0.002
12	6	4,096	216,769	8,191	0.465	0.001
14	14	16,384	54,944,557	32,767	60.833	0.007
14	7	16,384	1,304,769	32,767	0.907	0.004

Both produce ADDs agreeing with the table on all 2^k instantiations, hence by canonicity the same ADD. For the unstructured factors Method 1's step count grows by a factor near 16 each time N quadruples — the predicted quadratic — while Method 2's is exactly $2N-1$ throughout, the $\Theta(N)$ of the hashing variant; the times track the counts, the gap reaching four orders of magnitude at $k = 14$. Local structure cuts Method 1's cost by about 42× at $k = 14$ but not its growth rate, there still being one APPLY per row at a cost proportional to the accumulator, while Method 2's cost is *independent* of local structure, the tree being built in full before anything merges.

Problem 13.13. Construct an ADD over variables A and B, which maps each instantiation to 1 if either A or B is true and to 0 otherwise. Multiply this ADD with the one constructed in Exercise 13.8.

The ADD of Exercise 13.8 represents the CPT $\Pr(D \mid A, B, C)$ of Figure 13.21 under the order A, B, C, D; its reduced form is: root α labeled A with $\text{low}(\alpha) = \beta_2$, $\text{high}(\alpha) = \beta_1$; β_1 labeled B with $\text{low}(\beta_1) = \gamma$, $\text{high}(\beta_1) = \delta_1$; β_2 labeled B with $\text{low}(\beta_2) = \delta_3$, $\text{high}(\beta_2) = s_{.5}$; γ labeled C with $\text{low}(\gamma) = \delta_2$, $\text{high}(\gamma) = \delta_1$; and the three D-nodes $\delta_1 = (s_{.1}, s_{.9})$, $\delta_2 = (s_{.8}, s_{.2})$, $\delta_3 = (s_1, s_0)$, each written as (low, high).

Solution. The disjunction ADD. For $g(A, B) = 1$ iff $A \vee B$, under the same order $A \prec B \prec C \prec D$: A true makes the disjunction hold regardless of B, so the root's high child is the sink 1, and A false leaves the value of B. Hence

node	label	low child	high child
α' (root)	A	β'	t_1
β'	B	t_0	t_1

with sinks $t_0 = 0$, $t_1 = 1$; four nodes, reduced (Check!).

The product. Run APPLY (Algorithm 37) with $\odot = *$ on $\varphi_1 = \alpha$ and $\varphi_2 = \alpha'$. Both roots are labelled A, so line 6 recurses on the two branches.

- (i) A true: Apply$(\beta_1, t_1, *)$. Line 10 carries the sink t_1 unchanged into every branch of β_1 down to the sinks, where each value is multiplied by 1; since `unique_node` and `unique_sink` return the existing node whenever one matches, the branch rebuilds β_1 itself.
- (ii) A false: Apply$(\beta_2, \beta', *)$, both labelled B. On the high side $.5 \cdot 1 = .5$; on the low side Apply$(\delta_3, t_0, *)$ carries t_0 into both children of δ_3, giving 0 twice, so line 13 returns the sink 0 instead of creating a node — the whole δ_3 subgraph collapses. The two results differ, so a B-node $\beta_3 = (s_0, s_{.5})$ is created.

The two A-branches differ, so the root is an A-node with children β_3, β_1, and the reduced ADD of $h = f \cdot g$ is

node	label	low child (dotted)	high child (solid)
α_h (root)	A	β_3	β_1
β_1	B	γ	δ_1
β_3	B	s_0	$s_{.5}$
γ	C	δ_2	δ_1
δ_1	D	$s_{.1}$	$s_{.9}$
δ_2	D	$s_{.8}$	$s_{.2}$

with sinks $s_0, s_{.1}, s_{.2}, s_{.5}, s_{.8}, s_{.9}$: 12 nodes against the 14 of Exercise 13.8, the node δ_3 and the sink 1 having gone and β_2 having become β_3. Multiplying by g leaves f untouched where $A \vee B$ holds and zeroes the four rows where both are false:

A	B	C	D	f	g	$f \cdot g$
true	true	true	true	.9	1	.9
true	true	true	false	.1	1	.1
true	true	false	true	.9	1	.9
true	true	false	false	.1	1	.1
true	false	true	true	.9	1	.9
true	false	true	false	.1	1	.1
true	false	false	true	.2	1	.2
true	false	false	false	.8	1	.8
false	true	true	true	.5	1	.5
false	true	true	false	.5	1	.5
false	true	false	true	.5	1	.5
false	true	false	false	.5	1	.5
false	false	true	true	0	0	0
false	false	true	false	1	0	0
false	false	false	true	0	0	0
false	false	false	false	1	0	0

and tracing the sixteen rows through the ADD reproduces the last column (Check!).

Problem 13.14. Sum out variable B from the ADD constructed in Exercise 13.8.
That ADD represents the CPT $\Pr(D \mid A, B, C)$ of Figure 13.21 under the order A, B, C, D. Its reduced form is: root α labeled A with $\text{low}(\alpha) = \beta_2$, $\text{high}(\alpha) = \beta_1$; β_1 labeled B with $\text{low}(\beta_1) = \gamma$, $\text{high}(\beta_1) = \delta_1$; β_2 labeled B with $\text{low}(\beta_2) = \delta_3$, $\text{high}(\beta_2) = s_{.5}$; γ labeled C with $\text{low}(\gamma) = \delta_2$, $\text{high}(\gamma) = \delta_1$; and the D-nodes $\delta_1 = (s_{.1}, s_{.9})$, $\delta_2 = (s_{.8}, s_{.2})$, $\delta_3 = (s_1, s_0)$, written as (low, high).

Solution. By the method of Section 13.5.2 (Figure 13.16), summing out a variable is done by adding two restrictions:

$$\sum_B f \;=\; f^{B=\text{true}} \;+\; f^{B=\text{false}},$$

each restriction by RESTRICT (Algorithm 38), the addition by APPLY (Algorithm 37) with $\odot = +$.
Restrictions. RESTRICT follows the high child at a B-node (line 6) and recurses on both children elsewhere (lines 10–12). From α: $\beta_1 \mapsto \delta_1$ and $\beta_2 \mapsto s_{.5}$, which differ, so $f^{B=\text{true}}$ is the A-node with low $s_{.5}$ and high δ_1 — mentioning no C, as the CPT's B true half indeed does not. Taking low children instead, $\beta_1 \mapsto \gamma$ and $\beta_2 \mapsto \delta_3$, so $f^{B=\text{false}}$ is the A-node with low δ_3 and high γ.
Addition. Both roots are labelled A, so APPLY recurses on matching branches.

- (i) A true: add δ_1 to γ; since $C \prec D$, line 10 carries δ_1 into both branches of γ. Under C true, $\text{Apply}(\delta_1, \delta_1, +)$ gives $\varepsilon_1 = (s_{.2}, s_{1.8})$; under C false, $\text{Apply}(\delta_2, \delta_1, +)$ gives $\varepsilon_2 = (s_{.9}, s_{1.1})$. These differ, so a C-node $\gamma' = (\varepsilon_2, \varepsilon_1)$ is created.
- (ii) A false: add $s_{.5}$ to δ_3; the sink follows every variable, so line 10 applies with δ_3 first, giving $\varepsilon_3 = (s_{1.5}, s_{.5})$, independent of C.

The two A-branches differ, so the reduced ADD for $\sum_B f$, over A, C, D, is

node	label	low child (dotted)	high child (solid)
α'' (root)	A	ε_3	γ'
γ'	C	ε_2	ε_1
ε_1	D	$s_{.2}$	$s_{1.8}$
ε_2	D	$s_{.9}$	$s_{1.1}$
ε_3	D	$s_{1.5}$	$s_{.5}$

with the six sinks $s_{.2}, s_{1.8}, s_{.9}, s_{1.1}, s_{1.5}, s_{.5}$: 11 nodes, reduced, since all six sink values are distinct, $\varepsilon_1 \neq \varepsilon_2$, and the three D-nodes have pairwise different child pairs. Summing out B from Figure 13.21 by hand,

A	C	D	$B = \text{true}$	$B = \text{false}$	$\sum_B$
true	true	true	.9	.9	1.8
true	true	false	.1	.1	.2
true	false	true	.9	.2	1.1
true	false	false	.1	.8	.9
false	true	true	.5	0	.5
false	true	false	.5	1	1.5
false	false	true	.5	0	.5
false	false	false	.5	1	1.5

and tracing the eight rows through the ADD reproduces the last column, the A false rows going straight to ε_3 and so not depending on C.

12.3 Exercises 13.15–13.21

Problem 13.15. Restrict the ADD constructed in Exercise 13.8 to $C = \text{false}$.
Recall that Exercise 13.8 asks for an ADD representation of the CPT $\Pr(D \mid A, B, C)$ of Figure 13.21, using the ADD variable order A, B, C, D. All four variables are binary and the CPT is:

A	B	C	D	$\Pr(D \mid A, B, C)$
true	true	true	true	.9
true	true	true	false	.1
true	true	false	true	.9
true	true	false	false	.1
true	false	true	true	.9
true	false	true	false	.1
true	false	false	true	.2
true	false	false	false	.8
false	true	true	true	.5
false	true	true	false	.5
false	true	false	true	.5
false	true	false	false	.5
false	false	true	true	0
false	false	true	false	1
false	false	false	true	0
false	false	false	false	1

The CPT is viewed here as a factor $f(A, B, C, D)$ over four binary variables. Apply the RESTRICT operation of Algorithm 38 with the instantiation $C = \text{false}$ and report the resulting reduced ADD.

Solution. Restricting to $C = \text{false}$ deletes the single C-node and hands its low child up. Exercise 13.8's reduced ADD φ under $A \prec B \prec C \prec D$, with D-nodes written as (high, low) pairs $\alpha = (.9, .1)$, $\beta = (.2, .8)$, $\delta = (0, 1)$ and the $(.5, .5)$ node collapsed to its sink, is:

node	label	high child	low child
r (root)	A	b_1	b_0
b_1	B	α	c_1
b_0	B	sink .5	δ
c_1	C	α	β
α	D	sink .9	sink .1
β	D	sink .2	sink .8
δ	D	sink 0	sink 1

with sinks $.9, .1, .2, .8, .5, 0, 1$: 14 nodes against 16 table rows, and C mentioned on only one path, recording that $\Pr(D \mid A, B, C) = \Pr(D \mid A, B)$ outside the context A true, B false.
RESTRICT (Algorithm 38) walks top down, descending into the child selected by the evidence at an instantiated node and rebuilding every other node, collapsing it when the two returned children coincide. Only C is instantiated, to false, so only c_1 is affected: it returns its low child β (line 8), whence

b_1 becomes $b_1' = (\alpha, \beta)$, which survives since $\alpha \neq \beta$, while $b_0, \alpha, \beta, \delta$ and the sinks are unchanged. Thus $\varphi^{C=\text{false}}$ is

node	label	high child	low child
r' (root)	A	b_1'	b_0
b_1'	B	α	β
b_0	B	sink .5	δ
α	D	sink .9	sink .1
β	D	sink .2	sink .8
δ	D	sink 0	sink 1

with the same seven sinks: 6 nonsink nodes, reduced (Check!), respecting $A \prec B \prec D$. It represents the factor over A, B, D obtained by deleting every $C = \text{true}$ row of the CPT:

A	B	D	value
true	true	true	.9
true	true	false	.1
true	false	true	.2
true	false	false	.8
false	true	true	.5
false	true	false	.5
false	false	true	0
false	false	false	1

For instance A true, B false, D true traces $r' \to b_1' \to \beta \to$ the sink .2, the CPT entry for A true, B false, C false, D true; the other seven rows likewise (Check!).

Problem 13.16. Let $f_1(\mathbf{X}_1)$ and $f_2(\mathbf{X}_2)$ be two factors and let φ_1 and φ_2 be their corresponding ADD representations using the same variable ordering. Show that the size of the ADD returned by $\text{APPLY}(\varphi_1, \varphi_2, \odot)$ is $O(\exp(|\mathbf{X}_1 \cup \mathbf{X}_2|))$.

Solution. With $\mathbf{Z} = \mathbf{X}_1 \cup \mathbf{X}_2$, $k = |\mathbf{Z}|$ and the common order restricted to $\mathbf{Z}$ written $V_1 \prec \cdots \prec V_k$, the output $\varphi = \text{APPLY}(\varphi_1, \varphi_2, \odot)$ is a reduced ADD over $\mathbf{Z}$ respecting $\prec$, and any such ADD has at most $2^{k+1} - 1$ nodes.

(i) *Only* $\mathbf{Z}$*-variables, in order.* In Algorithm 37 a nonsink node is created only by unique_node(l, h, V) on lines 9 and 13, with $V = \text{var}(\varphi_1)$ for arguments that are descendants of the original φ_1, φ_2, hence $V \in \mathbf{Z}$. The order is respected: on lines 7–8 both recursive arguments lie strictly below $\text{var}(\varphi_1)$, and on lines 11–12 line 1 with the line-10 guard forces $\text{var}(\varphi_1) \prec \text{var}(\varphi_2)$, so again both lie strictly below.

(ii) *Reduced, and every counted node reachable.* Lines 9 and 13 return l whenever $l = h$, so no node has two identical children, and unique_node/unique_sink return an existing node whenever one matches, so no two nodes share a label and child pair. An ADD is the rooted DAG at its root, so nodes built during the computation and then orphaned by a later collapse do not count towards $|\varphi|$.

(iii) *The counting bound.* Let ψ be an ADD over $V_1 \prec \cdots \prec V_k$ with every node reachable, and let u be labelled V_i. A root-to-u path π has every node above u labelled strictly before V_i, so π fixes a partial instantiation of $V_1, \ldots, V_{i-1}$; extending it arbitrarily to a full instantiation $\mathbf{v}$ and following $\mathbf{v}$ from the root reaches exactly u, since at each node on the way $\mathbf{v}$ selects the child lying on π. Hence $\mathbf{v} \mapsto$ the first node reached whose variable is not among $V_1, \ldots, V_{i-1}$ is onto the V_i-nodes, so

$$\#\{\text{nodes labeled } V_i\} \;\le\; 2^{\,i-1},$$

and the same argument on sinks, regarded as labelled by a variable following all others, gives $\#\{\text{sinks}\} \le 2^k$. Summing,

$$\begin{aligned} |\psi| \;&\le\; \sum_{i=1}^{k} 2^{\,i-1} \;+\; 2^k \\ &=\; (2^k - 1) + 2^k \;=\; 2^{k+1} - 1. \end{aligned}$$

Applying this to φ,

$$|\varphi| \;\le\; 2^{\,|\mathbf{X}_1 \cup \mathbf{X}_2|+1} - 1 \;=\; O(\exp(|\mathbf{X}_1 \cup \mathbf{X}_2|)).$$

Problem 13.17. We presented an $O(n^2)$ algorithm for summing out a variable from an ADD, where n is the size of the given ADD. The algorithm restricts the ADD twice and then adds the two results,

$$\sum_X f \;=\; \Big(\sum_X f^{X=\text{false}}\Big) + \Big(\sum_X f^{X=\text{true}}\Big),$$

using RESTRICT (Algorithm 38) twice and APPLY (Algorithm 37) once with $\odot$ taken to be numeric addition. Assuming that the summed-out variable appears last in the ADD variable order, show how the sum-out operation can be implemented in $O(n)$. Show that this complexity applies to summing out multiple variables as long as they appear last in the ADD order.

Solution. One bottom-up sweep does it, the summed-out variables occupying the bottom layers so that no APPLY of two large ADDs is ever needed. Throughout, all variables are binary, φ has n nodes (sinks included) and respects $V_1 \prec \cdots \prec V_k$, and a unique-table operation or an arithmetic operation on sink labels costs $O(1)$ — the accounting under which APPLY is $O(nm)$.

One variable, last in the order. With $X = V_k$, an X-node can only have sinks as children, so the X-nodes form the layer just above the sinks and one bottom-up sweep suffices — provided sinks reached on paths bypassing X are *doubled*: if the path of an instantiation $\mathbf{v}$ of $V_1, \ldots, V_{k-1}$ meets no X-node and ends at the sink s, then $f(\mathbf{v}, x) = s$ for both x, so $(\sum_X f)(\mathbf{v}) = 2s$. Process the nodes in reverse topological order, computing $g(u)$:

1. u a sink with label s: $g(u) \leftarrow$ unique_sink$(2s)$.
2. u labelled X, with sink children labelled a (low) and b (high): $g(u) \leftarrow$ unique_sink$(a + b)$ — reading the children's labels, *not* g of them.
3. u labelled V_i, $i < k$: with $l = g(\text{low}(u))$, $h = g(\text{high}(u))$, set $g(u) \leftarrow l$ if $l = h$ and $g(u) \leftarrow$ unique_node(l, h, V_i) otherwise.

Return $g(\text{root})$. Writing f_u for the factor at u, a bottom-up induction gives $g(u) = \sum_X f_u$ for every u not labelled X: a sink is constant in X, so $\sum_X f_u \equiv 2s$ as rule 1 builds; and at V_i with $i < k$, summing out commutes with the case split $f_u(V_i = \text{false}, \cdot) = f_{\text{low}(u)}$, $f_u(V_i = \text{true}, \cdot) = f_{\text{high}(u)}$, each child being covered by rule 2 (whose $a + b$ is the correct constant) or by the hypothesis, and rule 3 assembles them. The root is labelled X only when $k = 1$, which rule 2 handles. The output is reduced — rule 3 never builds a node with identical children and the unique table never duplicates — and respects $V_1 \prec \cdots \prec V_{k-1}$.

Each node is examined once for $O(1)$ work, and bucketing nodes by variable index and scanning from k down gives the reverse topological order in $O(n)$, so the total is $O(n)$ and the result has $O(n)$ nodes. In the general case the two restrictions are $O(n)$ each but the closing APPLY of two size-$O(n)$ ADDs is $O(n^2)$; that is the step the assumption removes, the two restrictions differing only in the layer above the sinks and so combining locally.

Several variables, a suffix of the order. Let $\mathbf{Y} = \{V_{k-m+1}, \ldots, V_k\}$, renamed $W_1 \prec \cdots \prec W_m$. By the ordering condition the sub-ADD at any $\mathbf{Y}$-labelled node mentions only $\mathbf{Y}$-variables, so φ splits into an upper part over $V_1, \ldots, V_{k-m}$ and maximal lower sub-ADDs over $\mathbf{Y}$, each of which summing out replaces by a single number. Put $\text{lvl}(u) = j$ for u labelled W_j and $m + 1$ for a sink, and for $\text{lvl}(u) = j$ let

$$\sigma(u) \;=\; \sum_{W_j, \ldots, W_m} f_u,$$

the total of f_u over the $\mathbf{Y}$-variables at or after W_j (an empty sum-range for a sink, so $\sigma(\text{sink}) =$ its label). If u is labeled W_j and c is one of its children with $\text{lvl}(c) = j'$, then f_c is constant in $W_{j+1}, \ldots, W_{j'-1}$, whence

$$\sum_{W_{j+1}, \ldots, W_m} f_c \;=\; 2^{j'-j-1}\, \sigma(c).$$

Therefore, bottom up,

$$\sigma(u) \;=\; 2^{j_l - j - 1}\, \sigma(\text{low}(u)) \;+\; 2^{j_h - j - 1}\, \sigma(\text{high}(u)),$$

where $j_l = \mathrm{lvl}(\mathrm{low}(u))$ and $j_h = \mathrm{lvl}(\mathrm{high}(u))$. Each $\sigma(u)$ costs $O(1)$ once the powers $2^0, 2^1, \ldots, 2^m$ have been tabulated, which is $O(m) = O(n)$ work done once.
Now define g on all nodes:

1. If $\mathrm{lvl}(u) \le m+1$, i.e. u is a sink or is labeled with a $\mathbf{Y}$-variable, set $g(u) \leftarrow \mathrm{unique\_sink}\big(2^{\mathrm{lvl}(u)-1}\,\sigma(u)\big)$. The factor $2^{\mathrm{lvl}(u)-1}$ accounts for the $\mathbf{Y}$-variables $W_1, \ldots, W_{\mathrm{lvl}(u)-1}$ that the path skipped, in which f_u is constant.
2. If u is labeled V_i with $i \le k-m$, set $l = g(\mathrm{low}(u))$, $h = g(\mathrm{high}(u))$, and $g(u) \leftarrow l$ if $l = h$, otherwise $g(u) \leftarrow \mathrm{unique\_node}(l, h, V_i)$.

Return $g(\mathrm{root})$.
Correctness is the same induction as before: for a node u above the $\mathbf{Y}$-layers, $g(u)$ represents $\sum_{\mathbf{Y}} f_u$, because summing out commutes with the case split at u and because rule 1 supplies precisely $\sum_{\mathbf{Y}} f_c$ for a child c in the $\mathbf{Y}$ region; the one-variable case is $m = 1$, where σ of an X-node is $a + b$ at $\mathrm{lvl} = 1$ and σ of a sink is s at $\mathrm{lvl} = 2$, giving $2s$. Every node is touched once for $O(1)$ arithmetic and $O(1)$ unique-table work, plus the $O(m)$ table of powers and the $O(n)$ bucket sort by level, so eliminating all m variables costs $O(n)$ altogether, not $O(mn)$.

Problem 13.18. Show how we can represent a factor over multivalued variables using an ADD with binary variables. Explain how the sum-out operation (of a multivalued variable) should be implemented in this case.

Solution. Binary-encode each value and pad with zero. For $f(X_1, \ldots, X_N)$ with $|X_t| = k_t$, take $b_t = \lceil \log_2 k_t \rceil$ binary ADD variables $X_t^1, \ldots, X_t^{b_t}$ and an injection c_t of X_t's values into $\{\mathrm{false}, \mathrm{true}\}^{b_t}$; a bit vector is **valid** if it lies in the image of c_t and **spurious** otherwise. Set

$$\hat{f}(\mathbf{v}) \;=\; \begin{cases} f(\mathbf{x}), & \mathbf{v} \text{ restricted to } X_t\text{'s bits is } c_t(x_t) \text{ for all } t, \\ 0, & \text{some } X_t\text{'s bits are spurious in } \mathbf{v}, \end{cases}$$

and represent f by the reduced ADD of $\hat{f}$ under any order keeping each variable's bits consecutive.
Zero padding is what makes both factor operations survive. Multiplication is unchanged: $\mathrm{APPLY}(\hat{f}_1, \hat{f}_2, \cdot)$ agrees with $f_1 f_2$ on valid instantiations and is $0 \cdot 0 = 0$ on spurious ones, so the product is again zero-padded. Sum-out of X_t is the sum-out of its b_t bits in succession,

$$\widehat{\textstyle\sum_{X_t} f} \;=\; \sum_{X_t^1} \cdots \sum_{X_t^{b_t}} \hat{f},$$

each an ordinary binary ADD sum-out: fixing the other bits at $\mathbf{w}$, the right-hand side is $\sum_{\mathbf{v}} \hat{f}(\mathbf{w}, \mathbf{v})$ over all 2^{b_t} vectors $\mathbf{v}$, the spurious ones contributing 0, so the value is 0 when $\mathbf{w}$ is spurious (as the encoding demands) and $\sum_{x_t} f(\mathbf{y}, x_t)$ when $\mathbf{w}$ encodes $\mathbf{y}$ – zero-padded again, so the invariant survives every elimination step. With X_t's bits placed last and consecutively, Exercise 13.17 does all b_t sum-outs in one bottom-up sweep of time $O(n)$ in the ADD size n.

Problem 13.19. Consider Bayesian networks of the type given in Figure 13.4 and let $\mathbf{e}$ be some evidence on nodes S_i. Let m be the number of nodes S_i set to true by evidence $\mathbf{e}$ and let k be the number of network edges. Show that $\Pr(D_j \mid \mathbf{e})$ can be computed in $O(k \exp(m))$.
Figure 13.4(a) is a two-layer network. Its roots are the binary nodes D_1, D_2, D_3, D_4, each with its own prior; its leaves are the binary nodes S_1, S_2, S_3, and the edges are

$$\begin{aligned} &D_1 \to S_1, \quad D_2 \to S_1, \quad D_3 \to S_1, \\ &D_2 \to S_2, \quad D_3 \to S_2, \quad D_4 \to S_2, \\ &D_3 \to S_3, \quad D_4 \to S_3, \end{aligned}$$

so $\mathbf{Pa}(S_1) = \{D_1, D_2, D_3\}$, $\mathbf{Pa}(S_2) = \{D_2, D_3, D_4\}$ and $\mathbf{Pa}(S_3) = \{D_3, D_4\}$, giving $k = 8$ edges. Each leaf node S_i is a logical-or of its parents. Networks "of this type" are the general such two-layer

networks: binary roots $D_1, \dots, D_n$ with arbitrary priors, binary leaves $S_1, \dots, S_p$ whose parents are arbitrary subsets of the roots, each leaf being the logical-or of its parents, and $k = \sum_i |\mathbf{Pa}(S_i)|$.
Hint: prune the network and appeal to the inclusion-exclusion principle, which states that

$$\Pr(\alpha_1 \vee \cdots \vee \alpha_n) \;=\; \sum_{t=1}^{n} (-1)^{t-1} \sum_{I \subseteq \{1,\dots,n\},\, |I|=t} \Pr\Big(\bigwedge_{i \in I} \alpha_i\Big),$$

an equation with $2^n - 1$ terms.

Solution. Prune to an event over the roots, then expand the m positive observations by inclusion-exclusion; the resulting $(**)$ has 2^m terms of cost $O(k)$ each. Write $p_j = \Pr(D_j = \text{true})$, $\bar{p}_j = 1 - p_j$, let $\mathbf{e}$ set S_i false for $i \in N$ and true for $i \in T$ with $|T| = m$, and put $\alpha_i = \bigvee_{D \in \mathbf{Pa}(S_i)} (D = \text{true})$. Unobserved leaves are barren, hence deletable (Chapter 6), and the surviving leaf CPTs are indicators, $\Pr(s_i \mid \mathbf{d}) = [\mathbf{d} \models \alpha_i]$ and $\Pr(\bar{s}_i \mid \mathbf{d}) = [\mathbf{d} \models \neg\alpha_i]$, so $\Pr(\mathbf{e}) = \Pr(\bigwedge_{i \in N} \neg\alpha_i \wedge \bigwedge_{i \in T} \alpha_i)$, an event over the mutually independent roots alone (Section 13.2.3). Since $\neg\alpha_i$ says every parent of S_i is false,

$$\bigwedge_{i \in N} \neg\alpha_i \;\equiv\; \bigwedge_{D \in \mathbf{F}} (D = \text{false}), \qquad \mathbf{F} \;=\; \bigcup_{i \in N} \mathbf{Pa}(S_i).$$

found in $O(k)$ by one edge scan. Conditioning on it leaves the other roots at their priors and reduces each α_i, $i \in T$, to $\alpha_i' = \bigvee_{D \in \mathbf{Pa}(S_i) \setminus \mathbf{F}} (D = \text{true})$, whence $\Pr(\mathbf{e}) = (\prod_{D_j \in \mathbf{F}} \bar{p}_j) \Pr(\bigwedge_{i \in T} \alpha_i')$. Complementing the remaining conjunction and applying the hint,

$$\Pr\Big(\bigwedge_{i \in T} \alpha_i'\Big) = 1 - \Pr\Big(\bigvee_{i \in T} \neg\alpha_i'\Big) = \sum_{I \subseteq T} (-1)^{|I|} \Pr\Big(\bigwedge_{i \in I} \neg\alpha_i'\Big),$$

the $I = \emptyset$ term being 1, and each remaining term a conjunction of negative literals,

$$\bigwedge_{i \in I} \neg\alpha_i' \;\equiv\; \bigwedge_{D \in \mathbf{U}_I} (D = \text{false}), \qquad \mathbf{U}_I \;=\; \bigcup_{i \in I} \big(\mathbf{Pa}(S_i) \setminus \mathbf{F}\big),$$

of probability $\prod_{D_j \in \mathbf{U}_I} \bar{p}_j$ by independence of the roots. Hence

$$\Pr(\mathbf{e}) \;=\; \Big(\prod_{D_j \in \mathbf{F}} \bar{p}_j\Big) \sum_{I \subseteq T} (-1)^{|I|} \prod_{D_j \in \mathbf{U}_I} \bar{p}_j. \tag{$**$}$$

(Contradictory evidence – some α_{i_0}' empty – gives $\mathbf{U}_{I \cup \{i_0\}} = \mathbf{U}_I$, so terms cancel in pairs and $(**)$ returns 0.) For the joint, $\Pr(D_j = \text{true}, \mathbf{e}) = 0$ when $D_j \in \mathbf{F}$; otherwise clamping D_j true contributes p_j and satisfies α_i' for every $i \in T$ with $D_j \in \mathbf{Pa}(S_i)$, leaving $T_j = \{i \in T : D_j \notin \mathbf{Pa}(S_i)\}$, whose sets $\mathbf{U}_I$ never contain D_j:

$$\begin{aligned} \Pr(D_j = \text{true}, \mathbf{e}) \;&=\; p_j \Big(\prod_{D_l \in \mathbf{F}} \bar{p}_l\Big) \\ &\quad \times \sum_{I \subseteq T_j} (-1)^{|I|} \prod_{D_l \in \mathbf{U}_I} \bar{p}_l, \end{aligned}$$

with $\Pr(D_j \mid \mathbf{e})$ the ratio of the two. Each of the 2^m terms costs $O(k)$ – mark the parents of the S_i, $i \in I$, in a Boolean array indexed by roots and read $\prod_{D_j \in \mathbf{U}_I} \bar{p}_j$ off in one scan – so $\Pr(\mathbf{e})$ and the joint both cost $O(k \exp(m))$.

Problem 13.20. Consider Exercise 13.19. Show that the same complexity still holds if each node S_i is a noisy-or of its parents.
That is, the network is again two-layer: binary roots $D_1, \dots, D_n$ with priors $p_j = \Pr(D_j = \text{true})$, binary leaves $S_1, \dots, S_p$ with $\mathbf{Pa}(S_i) \subseteq \{D_1, \dots, D_n\}$ and $k = \sum_i |\mathbf{Pa}(S_i)|$ edges. But now each leaf

is a noisy-or (Section 5.4.1) with a suppressor probability $q_{ij} \in [0,1]$ for each edge $D_j \to S_i$ and a leak probability $l_i \in [0,1]$ for S_i, so that by Equation 5.2

$$\Pr(\bar{s}_i \mid \mathbf{d}) \;=\; (1 - l_i) \prod_{\substack{D_j \in \mathbf{Pa}(S_i) \\ d_j = \text{true}}} q_{ij}.$$

Let $\mathbf{e}$ be evidence on the nodes S_i and let m be the number of them set to true. Show that $\Pr(D_j \mid \mathbf{e})$ can still be computed in $O(k \exp(m))$.

Solution. Formula (†) below evaluates $\Pr(\mathbf{e})$ in 2^m terms of cost $O(k)$ each. Pruning is unavailable – a false noisy-or with nonzero suppressors does not force its parents false – but the property actually needed survives: $\Pr(\bar{s}_i \mid \mathbf{d})$ is a product over the parents, one factor per parent, each depending on that parent alone. As before $\mathbf{e}$ sets S_i false for $i \in N$ and true for $i \in T$ with $|T| = m$, unobserved leaves are barren, $\bar{p}_j = 1 - p_j$, and for a set J of leaves

$$Q_j(J) \;=\; \prod_{\substack{i \in J \\ D_j \in \mathbf{Pa}(S_i)}} q_{ij}$$

collects the suppressors on the edges from D_j into J (empty product $= 1$). The network factorisation gives $\Pr(\mathbf{e}) = \sum_{\mathbf{d}} \Pr(\mathbf{d}) \prod_{i \in N} \Pr(\bar{s}_i \mid \mathbf{d}) \prod_{i \in T} (1 - \Pr(\bar{s}_i \mid \mathbf{d}))$; expanding the last product term by term (inclusion-exclusion, as in Exercise 13.19) and interchanging the sums,

$$\Pr(\mathbf{e}) \;=\; \sum_{I \subseteq T} (-1)^{|I|}\, W(N \cup I), \qquad W(J) \;=\; \sum_{\mathbf{d}} \Pr(\mathbf{d}) \prod_{i \in J} \Pr(\bar{s}_i \mid \mathbf{d}).$$

By the noisy-or CPTs and root independence each $W(J)$ decomposes over the roots,

$$\begin{aligned} W(J) &= \sum_{\mathbf{d}} \prod_{j=1}^{n} \Pr(d_j) \prod_{i \in J} (1 - l_i) \prod_{\substack{D_j \in \mathbf{Pa}(S_i) \\ d_j = \text{true}}} q_{ij} \\ &= \Big(\prod_{i \in J} (1 - l_i) \Big) \sum_{\mathbf{d}} \prod_{j=1}^{n} \Pr(d_j)\, Q_j(J)^{[d_j = \text{true}]} \\ &= \Big(\prod_{i \in J} (1 - l_i) \Big) \prod_{j=1}^{n} \big(\bar{p}_j + p_j\, Q_j(J) \big). \end{aligned}$$

the second line regrouping the suppressors by root rather than by leaf, the third distributing (legitimate: the summand is a product of n terms, the j-th depending on d_j alone). Hence

$$\Pr(\mathbf{e}) \;=\; \sum_{I \subseteq T} (-1)^{|I|} \Big(\prod_{i \in N \cup I} (1 - l_i) \Big) \prod_{j=1}^{n} \big(\bar{p}_j + p_j\, Q_j(N \cup I) \big). \qquad (\dagger)$$

Clamping $D_j = \text{true}$ removes the sum over d_j and replaces the j-th bracket by its true half,

$$\begin{aligned} \Pr(D_j = \text{true}, \mathbf{e}) \;=\; \sum_{I \subseteq T} (-1)^{|I|} \Big(\prod_{i \in N \cup I} (1 - l_i) \Big) p_j\, Q_j(N \cup I) \\ \times \prod_{l \neq j} \big(\bar{p}_l + p_l\, Q_l(N \cup I) \big), \end{aligned}$$

and $\Pr(D_j \mid \mathbf{e})$ is the quotient by (†). Assuming as in Figure 13.4 that every leaf has a parent, so $m \le p \le k$: for fixed I and $J = N \cup I$, all the $Q_j(J)$ come from one $O(k)$ pass over the edges incident to J (and one to undo it), the leak factors number $|I| \le k$, and the outer product need only run over the at most k roots with a child, a childless root having $Q_j \equiv 1$ and bracket $\bar{p}_j + p_j = 1$. So each of the 2^m terms costs $O(k)$ and both quantities cost $O(k \exp(m))$.

Problem 13.21. Consider the problem of encoding equal parameters, as discussed in Section 13.3.2. Show that the following technique can be adopted to deal with this problem:

- Do not drop clauses for parameters with value 1.
- Replace each set of equal parameters θ_i in the same CPT by a new parameter η and drop all PI clauses of parameters θ_i.
- Add the following clauses $\neg\eta_i \vee \neg\eta_j$ for $i \neq j$ and $\neg\eta_i \vee \neg\theta_j$ for all i, j. Here $\eta_1, \ldots, \eta_k$ are all the newly introduced parameters to a CPT and $\theta_1, \ldots, \theta_m$ are all surviving old parameters in the CPT.

In particular, show that the old and new CNF encodings agree on their weighted model counts.

Solution. Both encodings have weighted model count $\Pr(\mathbf{e})$ under any evidence, because in each of them every model sets exactly one parameter variable per CPT true, namely the one carrying that CPT's parameter for the instantiation encoded by the indicators.

In the notation of Section 13.3.1, a CPT with family variables $F_1, \ldots, F_n$ and family instantiation $\mathbf{f}$ carries a variable $P_\mathbf{f}$ of weight $\theta_\mathbf{f}$, the IP clause $I_\mathbf{f} \Rightarrow P_\mathbf{f}$ and the PI clauses $\neg P_\mathbf{f} \vee I_{f_t}$, where $I_\mathbf{f} = I_{f_1} \wedge \cdots \wedge I_{f_n}$; all other literals weigh 1. Both Δ (Section 13.3.2) and Δ' replace a zero parameter by $\neg I_{f_1} \vee \cdots \vee \neg I_{f_n}$; Δ then drops value-1 parameters and encodes a merged group by $(I_{\mathbf{f}_1} \vee \cdots \vee I_{\mathbf{f}_m}) \iff \eta$, while Δ' gives each merged group G one η of the common weight (keeping each IP clause with $P_\mathbf{f}$ renamed to η, dropping the PI clauses of $\mathbf{f} \in G$), keeps every unmerged parameter with all its clauses – value-1 ones included – and adds $\neg\eta_i \vee \neg\eta_j$ $(i \neq j)$ and $\neg\eta_i \vee \neg P_{\mathbf{g}_j}$.

So each $\mathbf{f}$ with $\theta_\mathbf{f} \neq 0$ has exactly one variable $\nu(\mathbf{f})$ in Δ' – the η_i of its group, or $P_\mathbf{f}$ – of weight $\theta_\mathbf{f}$ and subject to $I_\mathbf{f} \Rightarrow \nu(\mathbf{f})$; this is what the first bullet buys, a dropped value-1 parameter leaving $\mathbf{f}$ with no variable at all.

Lemma. If $A \models \Delta'$ has indicators encoding the network instantiation $\mathbf{z}$, and $\mathbf{f}^\star$ is the family instantiation $\mathbf{z}$ induces in a given CPT, then $\theta_{\mathbf{f}^\star} \neq 0$ and A sets $\nu(\mathbf{f}^\star)$ true and every other parameter variable of that CPT false.

Were $\theta_{\mathbf{f}^\star} = 0$, the clause $\neg I_{f_1^\star} \vee \cdots \vee \neg I_{f_n^\star}$ would be falsified; so $\nu = \nu(\mathbf{f}^\star)$ exists and $I_{\mathbf{f}^\star} \Rightarrow \nu$ forces $A(\nu) = \text{true}$. Any surviving $P_{\mathbf{g}_j}$, $\mathbf{g}_j \neq \mathbf{f}^\star$, is false because its retained PI clauses imply an indicator of $\mathbf{g}_j$ that $\mathbf{z}$ falsifies – which is why no $\neg P_{\mathbf{g}_i} \vee \neg P_{\mathbf{g}_j}$ is needed. For the rest, by the third bullet (the merged parameters having no PI clauses, nothing else would stop a gratuitous η_i):

- (i) $\nu = \eta_{i_0}$: every other η_i is false by $\neg\eta_i \vee \neg\eta_{i_0}$, and every $P_{\mathbf{g}_j}$ by $\neg\eta_{i_0} \vee \neg P_{\mathbf{g}_j}$.
- (ii) $\nu = P_{\mathbf{f}^\star}$: every η_i is false by $\neg\eta_i \vee \neg P_{\mathbf{f}^\star}$.

For any evidence $\mathbf{e}$, models of Δ' then biject with the instantiations $\mathbf{z}$ having no zero parameter: the indicator clauses make A encode a unique $\mathbf{z}$, and conversely $A_\mathbf{z}$ – indicators per $\mathbf{z}$, $\nu(\mathbf{f}^\star)$ true in each CPT, all other parameter variables false – satisfies Δ':

- indicator clauses: immediate;
- zero-parameter clauses $\neg I_{f_1} \vee \cdots \vee \neg I_{f_n}$: here $\mathbf{f} \neq \mathbf{f}^\star$ because $\theta_\mathbf{f} = 0 \neq \theta_{\mathbf{f}^\star}$, so some indicator of $\mathbf{f}$ is false;
- IP clauses $I_\mathbf{f} \Rightarrow \nu(\mathbf{f})$: for $\mathbf{f} \neq \mathbf{f}^\star$ the antecedent is false; for $\mathbf{f} = \mathbf{f}^\star$ the consequent is true;
- PI clauses of surviving old parameters: the only such variable set true is $P_{\mathbf{f}^\star}$ (when $\mathbf{f}^\star$ was not merged), and all indicators of $\mathbf{f}^\star$ are true;
- the added binary clauses: at most one parameter variable per CPT is true.

and by the Lemma it is the only such model. The evidence literals restrict to the $\mathbf{z}$ compatible with $\mathbf{e}$, and since indicator literals and negative parameter literals weigh 1,

$$\text{weight}(A_\mathbf{z}) \;=\; \prod_{\text{CPTs}} \theta_{\mathbf{f}^\star} \;=\; \Pr(\mathbf{z}),$$

by the chain rule, and instantiations discarded for a zero parameter have $\Pr(\mathbf{z}) = 0$ and may be added back for free. Therefore

$$\text{WMC}\big(\Delta' \wedge I_{e_1} \wedge \cdots \wedge I_{e_r}\big) \;=\; \sum_{\mathbf{z} \sim \mathbf{e}} \Pr(\mathbf{z}) \;=\; \Pr(\mathbf{e}).$$

The same argument applied to Δ – where $(I_{\mathbf{f}_1} \vee \cdots \vee I_{\mathbf{f}_m}) \iff \eta$ forces η true exactly when $\mathbf{z}$ realises

a merged instantiation, and a dropped value-1 parameter contributes the factor 1 it should – gives $\mathrm{WMC}(\Delta \wedge I_{e_1} \wedge \cdots \wedge I_{e_r}) = \Pr(\mathbf{e})$ as well, so the two encodings agree.

12.4 Exercises 13.22–13.22

Problem 13.22. Consider the problem of encoding equal parameters as discussed in Section 13.3.2. Recall the CNF encoding of a Bayesian network from Sections 11.6.1 and 13.3.1. It uses an indicator variable I_x for each network variable X and value x, and a parameter variable $P_{x|\mathbf{u}}$ for each network parameter $\theta_{x|\mathbf{u}}$. Its clauses are the indicator clauses, which say that exactly one indicator of each variable X is true,

$$I_{x_1} \vee \ldots \vee I_{x_k}, \qquad \neg I_{x_i} \vee \neg I_{x_j} \ (i \neq j),$$

together with the parameter clauses, one group per parameter,

$$I_{u_1} \wedge \ldots \wedge I_{u_p} \wedge I_x \iff P_{x|\mathbf{u}}, \qquad \mathbf{u} = u_1, \ldots, u_p.$$

The left-to-right direction of the last equivalence is a single clause, $\neg I_{u_1} \vee \ldots \vee \neg I_{u_p} \vee \neg I_x \vee P_{x|\mathbf{u}}$, called the IP clause of the parameter; the right-to-left direction is the set of binary PI clauses $\neg P_{x|\mathbf{u}} \vee I_{u_i}$ and $\neg P_{x|\mathbf{u}} \vee I_x$. Literals I_x, $\neg I_x$ and $\neg P_{x|\mathbf{u}}$ have weight 1, and literal $P_{x|\mathbf{u}}$ has weight $\theta_{x|\mathbf{u}}$.
Consider the following technique for dealing with equal parameters:

- Do not drop clauses for parameters with value 1.
- Replace each set of equal parameters θ_i in the same CPT by a new parameter η and drop all the PI clauses of parameters θ_i.

Let f be the MLF encoded by the resulting CNF Δ. Recall from Section 12.4.3 that a CNF encodes an MLF by interpreting each of its models ω as the term containing exactly those variables that ω sets to true, and summing these terms; and recall from Exercise 12.20 that a term of f is minimal if the number of variables it contains is minimal among the terms of f. Show that the minimal terms of f correspond to the network polynomial. Hence, when coupled with the procedure of Exercise 12.20, this method can be used as an alternative for exploiting equal parameters.

Solution. Every complete instantiation forces exactly n class variables, so every term of f has size at least $2n$, and the terms of size exactly $2n$ are one per complete instantiation – the network polynomial with each parameter replaced by its class variable.
Let N have n variables with families X_iU_i, and $\mathbf{z}$ range over complete instantiations. Partition each CPT's parameters into classes of equal value, replacing each class of size ≥ 2 by a new variable (these are $\eta_1, \ldots, \eta_K$, collected in H) and leaving singletons with their old $P_{x|\mathbf{u}}$; write $c(x\mathbf{u})$ for the variable now representing $\theta_{x|\mathbf{u}}$ and $w(v)$ for the value a class variable v carries. So Δ is the indicator clauses, one IP clause $\neg I_{u_1} \vee \ldots \vee \neg I_{u_p} \vee \neg I_x \vee c(x\mathbf{u})$ per family instantiation – none missing, since by the first bullet a value-1 parameter still contributes one – and the PI clauses of the unmerged parameters only. A model ω sets exactly one indicator per network variable true, determining a complete instantiation $\mathbf{z}_\omega$, and conversely. Fixing $\mathbf{z}$, the IP clause of $x\mathbf{u}$ has all its negative literals falsified precisely when $x\mathbf{u} \sim \mathbf{z}$ and is otherwise already satisfied, so the IP clauses force exactly the variables in

$$E(\mathbf{z}) \;=\; \{\, c(x\mathbf{u}) \;:\; x\mathbf{u} \sim \mathbf{z} \,\},$$

and since classes are formed within a single CPT (distinct CPTs never share a class variable) while $\mathbf{z}$ is compatible with exactly one family instantiation of each of the n families, $|E(\mathbf{z})| = n$ for every $\mathbf{z}$ – the crux. The surviving PI clauses allow $P_{x|\mathbf{u}}$ true only when $x\mathbf{u} \sim \mathbf{z}$, i.e. only when it already lies in $E(\mathbf{z})$, so unmerged variables outside $E(\mathbf{z})$ are forced false and the merged ones in $H \setminus E(\mathbf{z})$ are free. Thus

$$\omega \;\longleftrightarrow\; (\mathbf{z}, S), \qquad \mathbf{z} \text{ a complete instantiation}, \quad E(\mathbf{z}) \subseteq S \subseteq E(\mathbf{z}) \cup H,$$

with S the parameter variables ω sets true. The term encoded by ω (Section 12.4.3) is

$$t_{\mathbf{z},S} \;=\; \Big(\prod_{x \sim \mathbf{z}} I_x\Big) \prod_{v \in S} v,$$

of size $n + |S|$, distinct models giving distinct terms. From $|E(\mathbf{z})| = n$ and $E(\mathbf{z}) \subseteq S$,

$$|t_{\mathbf{z},S}| \;=\; n + |S| \;\geq\; n + |E(\mathbf{z})| \;=\; 2n,$$

with equality iff $S = E(\mathbf{z})$, and $(\mathbf{z}, E(\mathbf{z}))$ is always a model (it satisfies every IP clause, and each unmerged variable it sets true is compatible with $\mathbf{z}$, hence satisfies its PI clauses). So the minimal terms of f are the terms of size $2n$, one per complete instantiation:

$$t_{\mathbf{z}} \;=\; \prod_{x \sim \mathbf{z}} I_x \prod_{x\mathbf{u} \sim \mathbf{z}} c(x\mathbf{u}).$$

By Definition 12.1 the network polynomial is

$$f_N \;=\; \sum_{\mathbf{z}} \prod_{\lambda_x \sim \mathbf{z}} \lambda_x \prod_{\theta_{x|\mathbf{u}} \sim \mathbf{z}} \theta_{x|\mathbf{u}},$$

one term per complete instantiation. Under the reading of Section 12.4.3 – $I_x \mapsto \lambda_x$, each positive parameter literal to the parameter it names – $t_{\mathbf{z}}$ becomes the term of $\mathbf{z}$ in f_N with every parameter replaced by its class variable (the polynomial of Exercise 12.19), and substituting $w(v)$ for each class variable recovers that term verbatim, since $c(x\mathbf{u})$ carries the value $\theta_{x|\mathbf{u}}$. Hence the minimal terms of f – equivalently f_m of Exercise 12.20 – correspond one-to-one to the terms of f_N, as required. (No term repeats a variable: classes live within a CPT and $\mathbf{z}$ selects one row of each.)

Coupling with Exercise 12.20: compile Δ into a decomposable, deterministic, smooth NNF Γ and run Exercise 2.13's pruning in dual form – 1 at positive-literal inputs, 0 at negative-literal and true inputs, ∞ at false, sum at and-nodes, minimum at or-nodes, then delete every or-edge to a child of different integer – whose argument applies verbatim with polarities exchanged, so Γ_m has as models exactly those of Γ setting fewest variables true (smoothness making this well defined) and inherits all three properties. By Exercise 12.17 the substitutions of Section 12.4.3 turn Γ_m into an arithmetic circuit for f_m, hence for the network polynomial with each equality class carried by its single η_j.

Illustration on Figure 13.1, with structure $A \to B$, $A \to C$, all variables binary, and CPTs

A	θ_A
true	.5
false	.5

A	B	$\theta_{B\|A}$
true	true	1
true	false	0
false	true	0
false	false	1

A	C	$\theta_{C\|A}$
true	true	.8
true	false	.2
false	true	.2
false	false	.8

Here $n = 3$, and the classes are $\eta_1 = \{\theta_{c|a}, \theta_{\bar{c}|\bar{a}}\}$ at .8, $\eta_2 = \{\theta_{\bar{c}|a}, \theta_{c|\bar{a}}\}$ at .2, $\eta_3 = \{\theta_a, \theta_{\bar{a}}\}$ at .5, $\eta_4 = \{\theta_{b|a}, \theta_{\bar{b}|\bar{a}}\}$ at 1, $\eta_5 = \{\theta_{\bar{b}|a}, \theta_{b|\bar{a}}\}$ at 0, so $K = 5$ and no PI clause survives. Besides the six indicator clauses, Δ has the ten IP clauses

$$\begin{array}{ll}
\neg I_a \vee \eta_3, & \neg I_{\bar{a}} \vee \eta_3, \\
\neg I_a \vee \neg I_b \vee \eta_4, & \neg I_{\bar{a}} \vee \neg I_{\bar{b}} \vee \eta_4, \\
\neg I_a \vee \neg I_{\bar{b}} \vee \eta_5, & \neg I_{\bar{a}} \vee \neg I_b \vee \eta_5, \\
\neg I_a \vee \neg I_c \vee \eta_1, & \neg I_{\bar{a}} \vee \neg I_{\bar{c}} \vee \eta_1, \\
\neg I_a \vee \neg I_{\bar{c}} \vee \eta_2, & \neg I_{\bar{a}} \vee \neg I_c \vee \eta_2.
\end{array}$$

So Δ has $8 \cdot 2^{5-3} = 32$ models and exactly 8 minimal terms, of size $2n = 6$, one per instantiation of A, B, C: $\mathbf{z} = a, b, c$ gives true variables $I_a, I_b, I_c, \eta_1, \eta_3, \eta_4$, hence $\lambda_a \lambda_b \lambda_c \, \eta_1 \eta_3 \eta_4 = .8 \cdot .5 \cdot 1 = \Pr(a, b, c)$ (Check!).

13 Approximate Inference by Belief Propagation

Contents

13.1 Exercises 14.1–14.7 220
13.2 Exercises 14.8–14.14 229
13.3 Exercises 14.15–14.20 237

13.1 Exercises 14.1–14.7

Problem 14.1. Consider the Bayesian network of Figure 14.17 and suppose that we condition on evidence $e : D = \text{true}$. Suppose that we have run Algorithm 40, IBP, on the network, where it converges and yields the (partial) set of messages and family marginals given in Figure 14.18.

Figure 14.17. The network has four binary variables. The roots are A and B; the leaves are C and D; the edges are $A \to C$, $B \to C$, $A \to D$, $B \to D$. Its CPTs are:

A	Θ_A
true	.5
false	.5

B	Θ_B
true	.25
false	.75

A	B	C	$\Theta_{C\|AB}$
true	true	true	.9
true	true	false	.1
true	false	true	.8
true	false	false	.2
false	true	true	.7
false	true	false	.3
false	false	true	.6
false	false	false	.4

A	B	D	$\Theta_{D\|AB}$
true	true	true	.6
true	true	false	.4
true	false	true	.4
true	false	false	.6
false	true	true	.3
false	true	false	.7
false	false	true	.3
false	false	false	.7

Figure 14.18 (partial IBP messages and marginal approximations; a dash marks a missing entry):

A	$\pi_C(A)$	$\pi_D(A)$	$\lambda_C(A)$	$\lambda_D(A)$
true	–	.5	.5	.6
false	–	.5	.5	.4

B	$\pi_C(B)$	$\pi_D(B)$	$\lambda_C(B)$	$\lambda_D(B)$
true	.3	.25	.5	–
false	.7	.75	.5	–

A	B	C	$BEL(ABC)$
true	true	true	.162
true	true	false	.018
true	false	true	–
true	false	false	.084
false	true	true	.084
false	true	false	.036
false	false	true	.168
false	false	false	.112

A	B	D	$BEL(ABD)$
true	true	true	.2
true	true	false	0
true	false	true	–
true	false	false	0
false	true	true	.1
false	true	false	0
false	false	true	.3
false	false	false	0

(a) Fill in the missing values for IBP messages in Figure 14.18.
(b) Fill in the missing values for family marginals in Figure 14.18.
(c) Compute marginals $BEL(A)$ and $BEL(B)$ using the IBP messages in Figure 14.18 and those computed in (a).
(d) Compute marginals $BEL(A)$ and $BEL(B)$ by summing out the appropriate variables from the family marginals $BEL(ABC)$ as well as $BEL(ABD)$.
(e) Compute the joint marginal $BEL(AB)$ by summing out the appropriate variables from family marginals $BEL(ABC)$ as well as $BEL(ABD)$.
Are the marginals computed in (d) consistent? What about those computed in (e)?

Solution. The missing entries are $\pi_C(A) = (.6, .4)$, $\lambda_D(B) = (.5625, .4375)$, $BEL(atbfct) = .336$ and $BEL(atbfdt) = .4$; the node marginals agree across families, the joint marginals on AB do not. Write λ_e for the evidence indicator ($\lambda_e(D) = (1, 0)$, $\lambda_e \equiv 1$ elsewhere).
(a) By Algorithm 40 (line 10) at the root A, with $\Theta_A = (.5, .5)$ and $\lambda_D(A) = (.6, .4)$,

$$\pi_C(A) \;=\; \eta\,\lambda_e(A)\,\Theta_A\,\lambda_D(A) \;=\; \eta\,(.30,\ .20) \;=\; (.6,\ .4),$$

and the same equation returns the printed $\pi_D(A) = \eta\,\Theta_A\lambda_C(A) = (.5, .5)$. By line 7, $\lambda_D(B) = \eta\sum_{A,D}\lambda_e(D)\Theta_{D|AB}\pi_D(A)$, the indicator killing every $D =$ false row, so with $\pi_D(A) = (.5, .5)$,

$$\lambda_D(B = \text{true}) = \eta\,[\,.5(.6) + .5(.3)\,] = \eta\,(.45),$$
$$\lambda_D(B = \text{false}) = \eta\,[\,.5(.4) + .5(.3)\,] = \eta\,(.35).$$

i.e. $\lambda_D(B) = (.45, .35)/.80 = (.5625, .4375)$, which reproduces the printed $\pi_C(B) = \eta\,\Theta_B\lambda_D(B) = \eta(.140625, .328125) = (.3, .7)$. Completed:

A	$\pi_C(A)$	$\pi_D(A)$	$\lambda_C(A)$	$\lambda_D(A)$
true	.6	.5	.5	.6
false	.4	.5	.5	.4
B	$\pi_C(B)$	$\pi_D(B)$	$\lambda_C(B)$	$\lambda_D(B)$
true	.3	.25	.5	.5625
false	.7	.75	.5	.4375

(b) Line 14 of Algorithm 40 gives $BEL(ABC) = \eta\,\lambda_e(C)\Theta_{C|AB}\pi_C(A)\pi_C(B)$ (C has no children, hence no λ factors). The products $\pi_C(a)\pi_C(b)$ are $.18, .42, .12, .28$ for $ab = tt, tf, ft, ff$ and already sum to 1, so $\eta = 1$ and the missing entry is $.42 \times .8 = .336$, the printed entries following likewise (Check!). Similarly $BEL(ABD) = \eta\,\lambda_e(D)\Theta_{D|AB}\pi_D(A)\pi_D(B)$, all $D =$ false rows vanishing; with $\pi_D(A) = (.5, .5)$, $\pi_D(B) = (.25, .75)$ the unnormalized $D =$ true entries are

$$ab = tt : .5(.25)(.6) = .075, \quad ab = tf : .5(.75)(.4) = .150,$$
$$ab = ft : .5(.25)(.3) = .0375, \quad ab = ff : .5(.75)(.3) = .1125,$$

summing to .375, so the missing entry is $.150/.375 = .4$ and the printed $.2, .1, .3$ follow. Completed:

A	B	C	$BEL(ABC)$
true	true	true	.162
true	true	false	.018
true	false	true	.336
true	false	false	.084
false	true	true	.084
false	true	false	.036
false	false	true	.168
false	false	false	.112

A	B	D	$BEL(ABD)$
true	true	true	.2
true	true	false	0
true	false	true	.4
true	false	false	0
false	true	true	.1
false	true	false	0
false	false	true	.3
false	false	false	0

(c) A root's family is the variable itself, so line 14 gives

$$\begin{aligned} BEL(A) &= \eta\, \Theta_A \lambda_C(A)\lambda_D(A) = \eta\,(.15,\ .10) = (.6,\ .4), \\ BEL(B) &= \eta\, \Theta_B \lambda_C(B)\lambda_D(B) \\ &= \eta\,(.0703125,\ .1640625) = (.3,\ .7). \end{aligned}$$

(d) From $BEL(ABC)$:

$$\begin{aligned} BEL(A = \text{true}) &= .162 + .018 + .336 + .084 = .6, \\ BEL(A = \text{false}) &= .084 + .036 + .168 + .112 = .4, \\ BEL(B = \text{true}) &= .162 + .018 + .084 + .036 = .3, \\ BEL(B = \text{false}) &= .336 + .084 + .168 + .112 = .7. \end{aligned}$$

From $BEL(ABD)$:

$$\begin{aligned} BEL(A = \text{true}) &= .2 + .4 = .6, \quad BEL(A = \text{false}) = .1 + .3 = .4, \\ BEL(B = \text{true}) &= .2 + .1 = .3, \quad BEL(B = \text{false}) = .4 + .3 = .7. \end{aligned}$$

(e) Summing C out of $BEL(ABC)$ and D out of $BEL(ABD)$:

A	B	from $BEL(ABC)$	from $BEL(ABD)$
true	true	.18	.2
true	false	.42	.4
false	true	.12	.1
false	false	.28	.3

The marginals of (d) are consistent – both families and the messages of (c) give $BEL(A) = (.6, .4)$, $BEL(B) = (.3, .7)$ – as an IBP fixed point must be, by the constraints $\sum_{xu\sim y} \mu_{xu} = \mu_y$ of Theorem 14.2. Those of (e) are not: .18 against .2 at $A = B = \text{true}$. Theorem 14.2 constrains only single shared variables, never shared sets, so nothing forces the two projections onto AB to agree.

Problem 14.2. Consider the following function,

$$f(x) \;=\; \frac{-.21x + .42}{-.12x + .54}$$

and the problem of identifying a fixed point $x^\star$,

$$x^\star = f(x^\star)$$

using fixed point iteration. We start with some initial value x_0, then at a given iteration $t > 0$ we update our current value x_t using the value x_{t-1} from the previous iteration:

$$x_t = f(x_{t-1}).$$

Starting with $x_0 = .2$, we have $x_1 \approx .7326$ and $x_2 \approx .5887$. In this particular example, repeated iterations will approach the desired solution $x^\star$.
(a) Continue evaluating x_t for $t \geq 3$ until the four most significant digits stop changing.
Consider next the Bayesian network of Figure 14.19 and the problem of identifying a parameter θ defining the CPT $\Theta_{\hat{A}}$, such that

$$\theta = \Pr(A = \text{true} \mid e).$$

Figure 14.19. A network over three binary variables A, $\hat{A}$, B with edges $A \to B$ and $\hat{A} \to B$, under evidence $e : B = \text{true}$. Its CPTs are:

A	Θ_A
true	.7
false	.3

$\hat{A}$	$\Theta_{\hat{A}}$
true	θ
false	$1 - \theta$

A	$\hat{A}$	B	$\Theta_{B\|A\hat{A}}$
true	true	true	.3
true	true	false	.7
true	false	true	.6
true	false	false	.4
false	true	true	.7
false	true	false	.3
false	false	true	.4
false	false	false	.6

(b) Suppose we set $\theta = .2$. What is $\Pr(A = \text{true} \mid e)$? Suppose we now reset θ to $\Pr(A = \text{true} \mid e)$. What is the new value of $\Pr(A = \text{true} \mid e)$?
(c) Design a fixed point iterative method to find such a θ.

Solution. (a) The iterates freeze at $x_t = .6219$ from $t = 7$ on (four significant digits):

t	x_t
0	.2000
1	.7326
2	.5887
3	.6314
4	.6191
5	.6227
6	.6216
7	.6219
8	.6219
9	.6219

The limit is exact: $x = f(x)$ reads $4x^2 - 25x + 14 = 0$, and of the roots $(25 \pm \sqrt{401})/8$ only $x^\star = (25 - \sqrt{401})/8 \approx .621877$ lies in $[0, 1]$.

(b) With A and $\hat{A}$ independent roots and B their common child under $e : B = \text{true}$,

$$\begin{aligned}\Pr(A{=}\mathrm{t}, e) &= \Theta_A(\mathrm{t}) \sum_{\hat{a}} \Theta_{\hat{A}}(\hat{a})\, \theta_{b|\mathrm{t},\hat{a}} \\ &= .7\,[\,\theta(.3) + (1-\theta)(.6)\,] \\ &= .7\,(.6 - .3\theta) = .42 - .21\theta,\end{aligned}$$

and likewise $\Pr(A{=}\mathrm{f}, e) = .3(.4 + .3\theta) = .12 + .09\theta$. Adding, $\Pr(e) = .54 - .12\theta$, and therefore

$$\Pr(A = \text{true} \mid e) \;=\; \frac{.42 - .21\theta}{.54 - .12\theta} \;=\; \frac{-.21\theta + .42}{-.12\theta + .54} \;=\; f(\theta),$$

the function f of part (a). So $\theta = .2$ gives $f(.2) = .378/.516 \approx .7326$, and resetting $\theta = .7326$ gives $f(.7326) \approx .5887$.

(c) The requirement is the fixed-point equation $\theta = f(\theta)$, so iterate: start from any $\theta_0 \in [0,1]$ (ED BP initializes at .5); at step t run exact inference in Figure 14.19 with $\Theta_{\hat{A}}$ set to θ_{t-1} and put $\theta_t = \Pr_{\theta_{t-1}}(A = \text{true} \mid e)$; stop when $|\theta_t - \theta_{t-1}|$ is below threshold. This is $\theta_t = f(\theta_{t-1})$ and converges from every θ_0 to $\theta^\star \approx .6219$: since

$$f'(x) = \frac{(-.21)(.54) - (.42)(-.12)}{(-.12x + .54)^2} = \frac{-.063}{(-.12x + .54)^2},$$

we get $|f'| \le .063/.42^2 = .357 < 1$ on $[0,1]$ (denominator in $[.42, .54]$), and f, decreasing, maps $[0,1]$ into $[f(1), f(0)] = [.5, .7778]$, so Banach applies. The loop is lines 5–11 of Algorithm 42 with line 8 as the inference step.

Problem 14.3. Consider the modified belief propagation algorithm (with normalizing constants), applied to a polytree network N inducing distribution Pr. For an edge $U \to X$, let e^+_{UX} be the evidence instantiating nodes on the U-side of edge $U \to X$ and let e^-_{UX} be the evidence instantiating nodes on the X-side of the edge. Prove the following:

$$\pi_X(U) = \Pr(U \mid e^+_{UX}),$$

$$\lambda_X(U) = \eta \Pr(e^-_{UX} \mid U),$$

where η is a constant that normalizes message $\lambda_X(U)$.

Solution. Induction on the number of nodes on the relevant side of the edge, the inductive step a regrouping of the chain-rule factorization of that side. The normalized message equations of Section 14.2, for X with parents $U = U_1, \ldots, U_m$ and children $Y_1, \ldots, Y_n$, are

$$\lambda_X(U_i) = \eta \sum_{XU \setminus \{U_i\}} \lambda_e(X)\, \Theta_{X|U} \prod_{k \ne i} \pi_X(U_k) \prod_j \lambda_{Y_j}(X),$$

$$\pi_{Y_j}(X) = \eta \sum_{U} \lambda_e(X)\, \Theta_{X|U} \prod_i \pi_X(U_i) \prod_{k \ne j} \lambda_{Y_k}(X),$$

with λ_e the evidence indicator and each η normalizing to sum one. Call $\mathbf{V}$ **parent-closed** if every parent of every $V \in \mathbf{V}$ lies in $\mathbf{V}$, and parent-closed **modulo** $Z \notin \mathbf{V}$ if every such parent lies in $\mathbf{V} \cup \{Z\}$.

Lemma. If $\mathbf{V}$ is parent-closed modulo Z and no member of $\mathbf{V}$ is an ancestor of Z, then

$$\Pr(\mathbf{V} \mid Z) = \prod_{V \in \mathbf{V}} \Theta_{V|\mathbf{Pa}(V)}.$$

(If $\mathbf{V}$ is parent-closed outright, the identity holds without the conditioning and without the ancestor proviso.)

Proof. $\mathbf{V}$, $\{Z\}$, $\mathrm{An}(Z)$ are pairwise disjoint and $\mathbf{W} = \mathbf{V} \cup \{Z\} \cup \mathrm{An}(Z)$ is parent-closed, so summing the chain rule $\Pr(\mathbf{X}) = \prod_X \Theta_{X|\mathbf{Pa}(X)}$ over $\mathbf{X} \setminus \mathbf{W}$ gives $\Pr(\mathbf{W}) = \prod_{V \in \mathbf{W}} \Theta_{V|\mathbf{Pa}(V)}$ (each summed-out

variable is a non-parent of everything retained, so its CPT sums to one, innermost first). No factor of a $\mathbf{V}$-node mentions a variable of $\mathrm{An}(Z)$, parents of $\mathbf{V}$-nodes lying in $\mathbf{V} \cup \{Z\}$, so summing over $\mathrm{An}(Z)$ yields

$$\Pr(\mathbf{V}, Z) = \Big(\prod_{V \in \mathbf{V}} \Theta_{V|\mathbf{Pa}(V)} \Big) \Pr(Z),$$

and dividing by $\Pr(Z)$ gives the claim. □

Deleting $U \to X$ from the skeleton splits N into the U-side $\mathbf{V}$, parent-closed, and the X-side $\mathbf{M}$, parent-closed modulo U; no node of $\mathbf{M}$ is an ancestor of U, since such a directed path ends with an edge into U, cannot traverse $U \to X$, and would be a second undirected route between the sides. So the Lemma applies to $\mathbf{M}$ with $Z = U$, and to every piece below – each one side of a deleted edge, conditioned on the far endpoint. Here e^+_{UX} is the part of e in $\mathbf{V}$ and e^-_{UX} the part in $\mathbf{M}$, and both claims go by simultaneous induction on the size of the relevant side, legitimately: each message on the right of a message equation crosses an edge with a strictly smaller relevant side.

The π message. Let U have parents $W_1, \dots, W_m$ and children $X, Y_1, \dots, Y_n$. Removing U from $\mathbf{V}$ disconnects it, again by single-connectedness, into

$$\mathbf{V} \setminus \{U\} \;=\; \biguplus_i \mathbf{V}_i^W \;\uplus\; \biguplus_j \mathbf{V}_j^Y,$$

with $\mathbf{V}_i^W$ the W_i-side of $W_i \to U$ and $\mathbf{V}_j^Y$ the Y_j-side of $U \to Y_j$. By the Lemma on $\mathbf{V}$, grouping factors by piece,

$$\begin{aligned}
\Pr(U, e^+_{UX}) &= \sum_{\mathbf{V}\setminus\{U\}} \prod_{V \in \mathbf{V}} \lambda_e(V)\, \Theta_{V|\mathbf{Pa}(V)} \\
&= \lambda_e(U) \sum_W \Theta_{U|W} \prod_i \Big[\textstyle\sum_{\mathbf{V}_i^W \setminus \{W_i\}} \prod \lambda_e \Theta \Big] \prod_j \Big[\textstyle\sum_{\mathbf{V}_j^Y} \prod \lambda_e \Theta \Big] \\
&= \lambda_e(U) \sum_W \Theta_{U|W} \prod_i \Pr(W_i, e^+_{W_iU}) \prod_j \Pr(e^-_{UY_j} \mid U),
\end{aligned}$$

the last line by the Lemma on each piece, the interchange of sum and product legitimate because the pieces are disjoint and share only the fixed U. By the induction hypotheses $\pi_U(W_i) \propto \Pr(W_i, e^+_{W_iU})$ and $\lambda_{Y_j}(U) = \eta_j \Pr(e^-_{UY_j} \mid U)$, the message equation for $\pi_X(U)$ therefore returns $\eta' \Pr(U, e^+_{UX})$ with η' independent of U; normalization over U forces $\eta' = 1/\Pr(e^+_{UX})$ and

$$\pi_X(U) = \Pr(U \mid e^+_{UX}).$$

The base case $m = n = 0$ is the same display with empty products: $\eta \lambda_e(U) \Theta_U = \Pr(U \mid e_U)$, and $e^+_{UX} = e_U$ there.

The λ message. Fix $U_i \to X$, with X having parents $U_1, \dots, U_m$ and children $Y_1, \dots, Y_n$. Removing X decomposes $\mathbf{M}$ as

$$\mathbf{M} \setminus \{X\} \;=\; \biguplus_{k \neq i} \mathbf{M}^k \;\uplus\; \biguplus_j \mathbf{M}^j,$$

with $\mathbf{M}^k$ the U_k-side of $U_k \to X$ and $\mathbf{M}^j$ the Y_j-side of $X \to Y_j$. By the Lemma on $\mathbf{M}$, parent-closed modulo U_i,

$$\begin{aligned}
\Pr(e^-_{U_iX} \mid U_i) &= \sum_{\mathbf{M}} \prod_{V \in \mathbf{M}} \lambda_e(V)\, \Theta_{V|\mathbf{Pa}(V)} \\
&= \sum_{XU\setminus\{U_i\}} \lambda_e(X)\, \Theta_{X|U} \prod_{k \neq i} \Pr(U_k, e^+_{U_kX}) \prod_j \Pr(e^-_{XY_j} \mid X),
\end{aligned}$$

by the same grouping. Comparing with the equation for $\lambda_X(U_i)$, and using $\pi_X(U_k) \propto \Pr(U_k, e^+_{U_kX})$ and $\lambda_{Y_j}(X) \propto \Pr(e^-_{XY_j} \mid X)$ with constants independent of U_i,

$$\lambda_X(U_i) = \eta \, \Pr(e^-_{U_iX} \mid U_i)$$

for the η normalizing $\lambda_X(U_i)$, independent of U_i. The base case $m = 1$, $n = 0$ is direct: $\lambda_X(U_1) = \eta \sum_X \lambda_e(X) \Theta_{X|U_1} = \eta \Pr(e^-_{U_1X} \mid U_1)$. □

Problem 14.4. Consider a polytree network N with families XU that induces a distribution $\Pr(\mathbf{X})$.
(a) Prove that the distribution $\Pr(\mathbf{X} \mid e)$ factorizes as

$$\Pr(\mathbf{X} \mid e) \;=\; \prod_{XU} \frac{\Pr(XU \mid e)}{\prod_{U \in U} \Pr(U \mid e)}.$$

(b) Prove that the factorization of (a) is equivalent to

$$\Pr(\mathbf{X} \mid e) \;=\; \prod_{XU} \frac{\Pr(XU \mid e)}{\Pr(X \mid e)^{\,n_X}},$$

where n_X is the number of children that variable X has in network N.

Solution. (a) The family structure of a polytree is a jointree with single-parent separators, and (a) is then the jointree factorization. Let T have the variables of N as nodes, an undirected edge $X - Y$ per directed edge $X \to Y$, cluster $C_X = XU$ at node X and separator $S_{XY} = \{X\}$ at edge $X - Y$; T is a tree since N is a polytree, and it is a jointree in the sense of Definition 9.13:

- Every family XU sits in the cluster C_X.
- (Running intersection.) If $Z \in C_X \cap C_Y$ then Z is equal or adjacent to each of X, Y in the skeleton, so by single-connectedness the unique T-path from X to Y is $X - Z - Y$ (degenerate when $Z \in \{X, Y\}$); every cluster and separator on it contains Z, separators being singletons of their edges' endpoints.
- Hence $C_X \cap C_Y = \{X\}$ for an edge $X - Y$: any further shared variable would force a second path between X and Y.

Now

$$\Pr(\mathbf{X} \mid e) \;=\; \eta \prod_{XU} \lambda_e(X)\, \Theta_{X|U},$$

with each factor $\lambda_e(X)\Theta_{X|U}$ a function of the cluster $C_X = XU$ alone, so $\Pr(\mathbf{X} \mid e)$ factorizes over the clusters of T and Theorem 9.12 gives

$$\Pr(\mathbf{X} \mid e) \;=\; \frac{\prod_i \Pr(C_i \mid e)}{\prod_{i-j} \Pr(S_{ij} \mid e)},$$

the general form (14.5). Theorem 9.12 is stated for a network-induced distribution while $\Pr(\mathbf{X} \mid e)$ is a conditional, so run its induction in the generality used – $\Pr(\mathbf{X} \mid e) = \prod_i \psi_i(C_i)$ for nonnegative ψ_i on the clusters of a running-intersection tree, here $\psi_{C_X} = \lambda_e(X)\Theta_{X|U}$ with η folded in. One cluster is trivial; otherwise take a leaf ℓ with neighbour p, separator $S = S_{\ell p}$, $R = C_\ell \setminus S$. Running intersection puts no variable of R in another cluster, so the factorization makes R independent of $\mathbf{X} \setminus C_\ell$ given S, whence

$$\begin{aligned}\Pr(\mathbf{X} \mid e) &= \Pr(R \mid S, e)\, \Pr(\mathbf{X} \setminus R \mid e) \\ &= \frac{\Pr(C_\ell \mid e)}{\Pr(S \mid e)}\, \Pr(\mathbf{X} \setminus R \mid e).\end{aligned}$$

and summing R out leaves $\Pr(\mathbf{X} \setminus R \mid e)$ factorizing over T minus ℓ (the factor $\sum_R \psi_\ell$ is a function of $S \subseteq C_p$, absorbed into ψ_p), to which the hypothesis applies. Here the clusters are the families and the tree edge from $U \to X$ carries separator $\{U\}$, so grouping tree edges by their head X,

$$\prod_{i-j} \Pr(S_{ij} \mid e) \;=\; \prod_{XU} \prod_{U \in U} \Pr(U \mid e).$$

Substituting,

$$\begin{aligned}\Pr(\mathbf{X} \mid e) &= \frac{\prod_{XU} \Pr(XU \mid e)}{\prod_{XU} \prod_{U \in U} \Pr(U \mid e)} \\ &= \prod_{XU} \frac{\Pr(XU \mid e)}{\prod_{U \in U} \Pr(U \mid e)},\end{aligned}$$

as required – the form assumed for $\Pr'$ in Equation 14.3.
(b) Only the denominator's bookkeeping differs. Grouping it by parent rather than by family, $\Pr(Z \mid e)$ appears in the inner product of family XU exactly when X is a child of Z, hence n_Z times in all, so

$$\prod_{XU} \prod_{U \in U} \Pr(U \mid e) \;=\; \prod_{Z \in \mathbf{X}} \Pr(Z \mid e)^{n_Z},$$

leaves ($n_Z = 0$) contributing nothing. Each X heads one family, so pushing $\Pr(X \mid e)^{n_X}$ into that family's term,

$$\begin{aligned}\Pr(\mathbf{X} \mid e) &= \frac{\prod_{XU} \Pr(XU \mid e)}{\prod_X \Pr(X \mid e)^{n_X}} \\ &= \prod_{XU} \frac{\Pr(XU \mid e)}{\Pr(X \mid e)^{n_X}},\end{aligned}$$

which is (b). □

Problem 14.5. Consider Theorem 14.1. Let N and N' be two Bayesian networks over the same set of variables $\mathbf{X}$, possibly having different structures, inducing distributions Pr and $\Pr'$ respectively, and let e be evidence. Prove that

$$\log \Pr(e) \;\geq\; ENT'(\mathbf{X} \mid e) + \sum_{XU} AVG'\big(\log \lambda_e(X)\, \Theta_{X|U}\big).$$

Here XU ranges over the families of N, $ENT'(\mathbf{X} \mid e)$ is the entropy of $\Pr'(\mathbf{X} \mid e)$, and $AVG'(\log \lambda_e(X)\Theta_{X|U}) = \sum_{xu} \Pr'(xu \mid e) \log \lambda_e(x)\theta_{x|u}$. Thus, if a distribution $\Pr'(\mathbf{X} \mid e)$ is induced by a sufficiently simple Bayesian network N', we can compute a lower bound on the probability of evidence $\Pr(e)$.

Solution. Drop the nonnegative KL term from Theorem 14.1, whose identity uses nothing about $\Pr'$ beyond its being a distribution over $\mathbf{X}$ – in particular it covers the one induced by N', whatever that structure is. Indeed, the chain rule for N with the evidence indicators gives $\Pr(x, e) = \prod_{XU} \lambda_e(x)\theta_{x|u}$ for complete x, the indicator product being 1 exactly on the x compatible with e; hence $\log \Pr(x \mid e) = \sum_{XU} \log \lambda_e(x)\theta_{x|u} - \log \Pr(e)$ and

$$\begin{aligned}KL\big(\Pr'(\mathbf{X} \mid e), \Pr(\mathbf{X} \mid e)\big) &= \sum_x \Pr'(x \mid e) \log \frac{\Pr'(x \mid e)}{\Pr(x \mid e)} \\ &= -ENT'(\mathbf{X} \mid e) - \sum_{XU} AVG'\big(\log \lambda_e(X)\Theta_{X|U}\big) + \log \Pr(e),\end{aligned}$$

the last step summing over the completions of xu, legitimate because $\log \lambda_e(x)\theta_{x|u}$ depends on x only through xu. Since $KL \geq 0$ (Section 14.4.1), solving for $\log \Pr(e)$ and dropping that term,

$$\log \Pr(e) \;\geq\; ENT'(\mathbf{X} \mid e) + \sum_{XU} AVG'\big(\log \lambda_e(X)\Theta_{X|U}\big),$$

as required. □

Problem 14.6. Consider running Algorithm 40, IBP, on a network N that does not have any evidence. Further, suppose that all IBP messages are initialized uniformly.
(a) Show that all λ messages are neutral. That is, for any given message $\lambda_X(U)$, show that all values $\lambda_X(u)$ are the same.
(b) Argue that when there is no evidence, IBP need not propagate any λ messages.
(c) Design a sequential message passing schedule where IBP is guaranteed to converge in a single iteration.

Hint: It may be useful to analyze how Algorithm 42, ED BP, parameterizes a fully disconnected network when the original network N has no evidence.

Solution. With e empty every indicator is $\lambda_e(x) \equiv 1$, and the normalized equations of Algorithm 40 (lines 7, 10, 14) make every π message sum to one.
(a) Induct on t; at $t = 0$ every $\lambda_X^0(U)$ is uniform. Suppose $\lambda_{Y_j}^{t-1}(X) \equiv c_j$ for all j, and let X have parents $U_1, \ldots, U_m$ and children $Y_1, \ldots, Y_n$. Line 7 reads

$$\lambda_X^t(U_i) = \eta \sum_{XU \setminus \{U_i\}} \lambda_e(X)\, \Theta_{X|U} \prod_{k \neq i} \pi_X^{t-1}(U_k) \prod_j \lambda_{Y_j}^{t-1}(X),$$

and the constants c_j come out of the sum, so for each value u_i,

$$\begin{aligned} \lambda_X^t(u_i) &= \eta\Big(\prod_j c_j\Big) \sum_x \sum_{u_{-i}} \theta_{x|u} \prod_{k \neq i} \pi_X^{t-1}(u_k) \\ &= \eta' \sum_{u_{-i}} \Big(\prod_{k \neq i} \pi_X^{t-1}(u_k)\Big) \underbrace{\sum_x \theta_{x|u}}_{=1} \\ &= \eta' \prod_{k \neq i} \underbrace{\Big(\sum_{u_k} \pi_X^{t-1}(u_k)\Big)}_{=1} \;=\; \eta', \end{aligned}$$

with u_{-i} the other parents, the two cancellations being that $\Theta_{X|U}$ is a conditional distribution and that the π messages are normalized. As η' does not depend on u_i, $\lambda_X^t(U_i)$ is neutral, and after normalization uniform again. □
(b) Each λ message being constant, wherever one appears in Algorithm 40 it contributes a multiplicative constant that η absorbs:

- In the π update (line 10), $\pi_{Y_j}^t(X) = \eta \sum_U \Theta_{X|U} \prod_i \pi_X^{t-1}(U_i) \prod_{k \neq j} \lambda_{Y_k}^{t-1}(X)$; the trailing product is a constant $\prod_{k \neq j} c_k$, so
$$\pi_{Y_j}^t(X) = \eta \sum_U \Theta_{X|U} \prod_i \pi_X^{t-1}(U_i),$$
independent of j, so every node sends the same π message to all its children.
- In the family marginal (line 14), $BEL(XU) = \eta\, \Theta_{X|U} \prod_i \pi_X(U_i) \prod_j \lambda_{Y_j}(X) = \eta\, \Theta_{X|U} \prod_i \pi_X(U_i)$, again with the λ factors absorbed.
- In the λ update itself, by (a) the output is uniform no matter what the inputs are.

No π message and no BEL therefore depends on the λ messages, which may be omitted: with no evidence IBP is pure causal propagation.
(c) Propagate only π messages, in topological order. Fix an order $X_1, \ldots, X_n$ of the variables with every parent before its children (N is acyclic); leave all λ messages at their uniform values; and for $r = 1, \ldots, n$ compute the single outgoing message

$$\pi(X_r) \;=\; \eta \sum_{U_r} \Theta_{X_r|U_r} \prod_{U \in U_r} \pi(U),$$

from the $\pi(U)$ computed earlier in the same sweep, sending it to every child of X_r (a root gets Θ_{X_r} from the empty product); finish with $BEL(X_r U_r) = \eta\, \Theta_{X_r|U_r} \prod_{U \in U_r} \pi(U)$. Writing $\pi(X)$ for the common value of the $\pi_{Y_j}(X)$ and ignoring λ messages is licensed by (b).
One sweep suffices: $\pi(X_r)$ depends only on the parent messages $\pi(U)$ and on the never-changing λ messages (through a constant η absorbs), and the $\pi(U)$ were computed strictly earlier, so by induction along the order recomputation returns the same value, while recomputing any λ message returns uniform by (a). Every message is stationary, the convergence test of line 3. (Sequential is essential – under the parallel schedule a root's influence takes as many iterations to reach a node as that node's depth.) In the ED BP picture of the hint, Theorem 14.6 identifies $\pi_X(U) = \Theta_{\hat{U}}$ and $\lambda_X(U) = \Theta_{\hat{s}|U}$, so (a) makes $\hat{S}$ independent of U, e' vacuous, and Equation 14.8 collapses to $\Theta_{\hat{U}} = \Pr'(U)$ – the one ancestral sweep above.

Problem 14.7. Let N and N' be two Bayesian networks over the same set of variables $\mathbf{X}$, where N' is fully disconnected. That is, N' induces the distribution

$$\Pr'(\mathbf{X}) \;=\; \prod_{X\in\mathbf{X}} \Theta'_X.$$

Identify CPTs Θ'_X for network N' that minimize $KL(\Pr, \Pr')$.

Solution. Take $\Theta'_X = \Pr(X)$ for every X; this is the unique minimizer. (The divergence here is weighted by Pr, the reverse of the direction in Theorem 14.1.) With $\mathbf{x}$ a complete instantiation, $\mathbf{x}[X]$ its value at X, and $\Pr'(\mathbf{x}) = \prod_X \theta'_{\mathbf{x}[X]}$,

$$\begin{aligned} KL(\Pr, \Pr') &= \sum_{\mathbf{x}} \Pr(\mathbf{x}) \log \Pr(\mathbf{x}) - \sum_{\mathbf{x}} \Pr(\mathbf{x}) \sum_{X} \log \theta'_{\mathbf{x}[X]} \\ &= -ENT(\mathbf{X}) \;-\; \sum_{X\in\mathbf{X}} \sum_{x} \Big(\sum_{\mathbf{x}\sim x} \Pr(\mathbf{x}) \Big) \log \theta'_x \\ &= -ENT(\mathbf{X}) \;-\; \sum_{X\in\mathbf{X}} \sum_{x} \Pr(x) \log \theta'_x, \end{aligned}$$

with $\mathbf{x} \sim x$ the completions of x. Here $ENT(\mathbf{X})$ is constant and the objective separates, each Θ'_X occurring in one term constrained only by $\sum_x \theta'_x = 1$, $\theta'_x \geq 0$. So maximize $\sum_x \Pr(x) \log \theta'_x$ per variable; against the candidate $\theta'_x = \Pr(x)$,

$$\begin{aligned} \sum_x \Pr(x) \log \theta'_x - \sum_x \Pr(x) \log \Pr(x) &= \sum_x \Pr(x) \log \frac{\theta'_x}{\Pr(x)} \\ &= -KL\big(\Pr(X), \Theta'_X\big) \;\leq\; 0 \end{aligned}$$

by nonnegativity of the KL divergence between the single-variable distributions $\Pr(X)$ and Θ'_X (Section 14.4.1), with equality iff $\Theta'_X = \Pr(X)$. So each subproblem has the unique maximizer $\theta'_x = \Pr(x)$, giving $\Pr'(\mathbf{X}) = \prod_X \Pr(X)$.

13.2 Exercises 14.8–14.14

Problem 14.8. Figure 14.6(a) depicts a Bayesian network $\mathcal{N}$ drawn as two three-by-two blocks of nodes joined by a single vertical edge. The upper block contains the labelled nodes A, B, U and three unlabelled nodes; the lower block contains the labelled nodes X, C and four unlabelled nodes. The edge $U \to X$ is the only edge of $\mathcal{N}$ joining the two blocks, so deleting it splits $\mathcal{N}$ into two disconnected pieces. The evidence is $\mathbf{e} = \bar{a}, b, \bar{c}$, so $\bar{a}, b$ lies on the U-side of the deleted edge and $\bar{c}$ lies on the X-side.

Figure 14.6(b) depicts the approximate network $\mathcal{N}'$ obtained by deleting $U \to X$: the edge is removed, a clone $\hat{U}$ (a root with CPT $\Theta_{\hat{U}}$, having the same values as U) is made the parent of X, and a binary node $\hat{S}$ with CPT $\Theta_{\hat{S}|U}$ is made a child of U and is assumed observed. The approximate evidence is $\mathbf{e}' = \bar{a}, b, \bar{c}, \hat{s}$.

For this network the edge-parameter conditions of Theorem 14.5,

$$\theta_{\hat{U}} = \Pr'(U \mid \mathbf{e}' - \hat{S}), \qquad \theta_{\hat{s}|U} = \eta \Pr'(\mathbf{e}' \mid \hat{U}),$$

(Equations 14.8 and 14.9) become

$$\theta_{\hat{U}} = \Pr'(U \mid \bar{a}, b, \bar{c}) \quad (14.10)$$

$$\theta_{\hat{s}|U} = \eta \Pr'(\bar{a}, b, \bar{c}, \hat{s} \mid \hat{U}). \quad (14.11)$$

Show that Equations 14.10 and 14.11 reduce to

$$\theta_{\hat{U}} = \Pr'(U \mid \bar{a}, b) \quad (14.12)$$

$$\theta_{\hat{s}|U} = \eta \Pr'(\bar{c} \mid \hat{U}). \quad (14.13)$$

Solution. Both reductions come from $\mathcal{N}'$ having two connected components. Deleting $U \to X$ severs the only link between the halves of $\mathcal{N}$, and the added nodes attach to opposite halves – $\hat{S}$ a child of U, $\hat{U}$ a root and parent of X – so

$$\mathcal{C}_U = \{A, B, U, \hat{S}, \ldots\}, \qquad \mathcal{C}_X = \{\hat{U}, X, C, \ldots\},$$

and the product of CPTs splits by component, making every set of $\mathcal{C}_U$-variables independent of every set of $\mathcal{C}_X$-variables under $\Pr'$.
Equation 14.10 to 14.12. Here $U, A, B \in \mathcal{C}_U$ and $C \in \mathcal{C}_X$, so $\Pr'(U, \bar{a}, b, \bar{c}) = \Pr'(U, \bar{a}, b) \Pr'(\bar{c})$; dividing by $\Pr'(\bar{a}, b, \bar{c}) = \Pr'(\bar{a}, b) \Pr'(\bar{c})$ and cancelling $\Pr'(\bar{c}) > 0$ gives $\Pr'(U \mid \bar{a}, b, \bar{c}) = \Pr'(U \mid \bar{a}, b)$, which is Equation 14.12.
Equation 14.11 to 14.13. Condition on a value $\hat{u}$ of the clone; $A, B, \hat{S} \in \mathcal{C}_U$ and $\hat{U} \in \mathcal{C}_X$, so splitting $\mathbf{e}'$ along the components,

$$\begin{aligned} \Pr'(\bar{a}, b, \bar{c}, \hat{s} \mid \hat{u}) &= \Pr'(\bar{a}, b, \hat{s} \mid \hat{u}) \cdot \Pr'(\bar{c} \mid \hat{u}) \\ &= \Pr'(\bar{a}, b, \hat{s}) \cdot \Pr'(\bar{c} \mid \hat{u}). \end{aligned}$$

the first equality by component independence, the second dropping a conditioning irrelevant to a $\mathcal{C}_U$-event. Since $\alpha = \Pr'(\bar{a}, b, \hat{s})$ is a positive constant independent of $\hat{u}$, and η in Equation 14.9 need only be positive, absorbing α into $\eta' = \eta\alpha > 0$ turns Equation 14.11 into $\theta_{\hat{s}|U} = \eta' \Pr'(\bar{c} \mid \hat{U})$, which is Equation 14.13. The constant's value is immaterial by Exercise 14.9, $\hat{S}$ being a leaf with the single parent U.

Problem 14.9. Let $\mathcal{N}$ be a Bayesian network that induces a distribution $\Pr(\mathbf{X})$ and let S be a leaf node in network $\mathcal{N}$ that has a single parent U. Show that the conditional distribution $\Pr(\mathbf{X} \setminus \{S\} \mid s)$ is the same for any two CPTs $\Theta_{S|U}$ and $\Theta'_{S|U}$ of node S as long as $\theta_{s|U} = \eta \cdot \theta'_{s|U}$ for some constant $\eta > 0$.

Solution. The CPT of S enters the slice $S = s$ only through the single factor $\theta_{s|u}$, so rescaling that column rescales the whole slice and cancels in the conditional. Write $\mathbf{Y} = \mathbf{X} \setminus \{S\}$ and let $\Pr_{-S}$ be induced by $\mathcal{N}$ with S deleted – a legitimate network over $\mathbf{Y}$, the leaf S occurring in no other family. By the chain rule, for every $\mathbf{y}$,

$$\Pr(\mathbf{y}, s) = \Big(\prod_{Z \in \mathbf{Y}} \theta_{z|\mathbf{u}_Z} \Big) \cdot \theta_{s|u(\mathbf{y})} = \Pr_{-S}(\mathbf{y}) \cdot \theta_{s|u(\mathbf{y})},$$

with $u(\mathbf{y})$ the value $\mathbf{y}$ assigns to U, the first factor running over the families of $\mathbf{Y}$ alone and hence independent of $\Theta_{S|U}$. Let $\Pr$, $\Pr'$ be induced by the two CPTs, all else fixed; they share $\Pr_{-S}$ and $\theta_{s|u} = \eta\theta'_{s|u}$ for every u, so $\Pr(\mathbf{y}, s) = \eta \Pr'(\mathbf{y}, s)$ for every $\mathbf{y}$. Summing gives $\Pr(s) = \eta \Pr'(s)$, positive since $\eta > 0$, and dividing,

$$\Pr(\mathbf{y} \mid s) = \frac{\eta \Pr'(\mathbf{y}, s)}{\eta \Pr'(s)} = \Pr'(\mathbf{y} \mid s). \qquad \square$$

Problem 14.10. Consider again Figure 14.7, which defines a network $\mathcal{N}$, and another network $\mathcal{N}'$ that results from deleting edge $A \to B$.
Network $\mathcal{N}$ of Figure 14.7(a) has nodes A, B, C, D and edges $A \to B$, $A \to C$, $B \to D$, $C \to D$.

Evidence is $\mathbf{e} : D = \text{true}$. Its CPTs are

A	θ_A
true	.8
false	.2

A	B	$\theta_{B\|A}$
true	true	.8
true	false	.2
false	true	.4
false	false	.6

A	C	$\theta_{C\|A}$
true	true	.5
true	false	.5
false	true	1.0
false	false	.0

B	C	D	$\theta_{D\|BC}$
true	true	true	.1
true	true	false	.9
true	false	true	.3
true	false	false	.7
false	true	true	.9
false	true	false	.1
false	false	true	.8
false	false	false	.2

Network $\mathcal{N}'$ of Figure 14.7(b) results from deleting $A \to B$: a clone $\hat{A}$ (a root) becomes the parent of B, with $\Theta_{B|\hat{A}}$ equal to the table $\Theta_{B|A}$ above, and an observed binary node $\hat{S}$ becomes a child of A. All other edges ($A \to C$, $B \to D$, $C \to D$) are retained, so $\mathcal{N}'$ is a polytree. The evidence is $\mathbf{e}' : D = \text{true}, \hat{S} = \text{true}$. Figure 14.8 depicts the edge parameters for $\mathcal{N}'$ computed using Algorithm 42, ED-BP, which converge after four iterations to

$\hat{A}$	$\theta_{\hat{A}}$
true	.8262
false	.1738

A	$\hat{S}$	$\theta_{\hat{S}\|A}$
true	true	.3438
true	false	.6562
false	true	.6562
false	false	.3438

Figure 14.20 depicts the messages computed by Algorithm 40, IBP, after converging on the original network $\mathcal{N}$:

Edge $A \to B$, then edge $A \to C$:

A	$\pi_B(A)$	$\lambda_B(A)$	$\pi_C(A)$	$\lambda_C(A)$
true	.8262	.3438	.6769	.5431
false	.1738	.6562	.3231	.4569

Edge $B \to D$:

B	$\pi_D(B)$	$\lambda_D(B)$
true	.7305	.1622
false	.2695	.8378

Edge $C \to D$:

C	$\pi_D(C)$	$\lambda_D(C)$
true	.6615	.4206
false	.3384	.5794

(a) Identify how edge parameters in $\mathcal{N}'$ correspond to IBP messages in $\mathcal{N}$.

(b) Consider the following correspondence between the approximations for family marginals com-

puted by IBP in $\mathcal{N}$ and by ED-BP in $\mathcal{N}'$:

A	B	$\mathrm{BEL}(AB) = \Pr'(\hat{A}B \mid \mathbf{e}')$
true	true	.3114
true	false	.4021
false	true	.0328
false	false	.2537

Compute $\mathrm{BEL}(A) = \Pr'(\hat{A} \mid \mathbf{e}')$.
(c) Given the results of (b), argue that we must have $\Pr'(\hat{A} \mid \mathbf{e}') = \Pr'(A \mid \mathbf{e}')$.
(d) Identify a pair of variables in $\mathcal{N}'$ whose joint marginal can be easily computed in $\mathcal{N}'$ but cannot be computed using IBP in $\mathcal{N}$.

Solution. (a) The two edge parameters of a deleted edge are the two IBP messages along it – Theorem 14.6 for full deletion, the discussion following it for any deletion leaving a polytree, as here:

$$\pi_X(U) = \theta_{\hat{U}}, \qquad \lambda_X(U) = \theta_{\hat{s}|U}.$$

The only deleted edge is $A \to B$, and the numbers confirm it:

$$\pi_B(A) = (.8262, .1738) = (\theta_{\hat{a}}, \theta_{\hat{\bar{a}}}),$$
$$\lambda_B(A) = (.3438, .6562) = (\theta_{\hat{s}|a}, \theta_{\hat{s}|\bar{a}}).$$

The messages on the retained edges have no counterpart among the edge parameters; their effect in $\mathcal{N}'$ comes from exact inference in the polytree.
(b) Sum B out of the family marginal:

$$\Pr'(\hat{A} = \text{true} \mid \mathbf{e}') = .3114 + .4021 = .7135,$$
$$\Pr'(\hat{A} = \text{false} \mid \mathbf{e}') = .0328 + .2537 = .2865.$$

So $\mathrm{BEL}(A) = (.7135, .2865)$, which is also the normalized $\pi_B(A)\lambda_B(A) = (.28405, .11405)$ that IBP computes on the edge $A \to B$.
(c) Theorem 14.6, in its polytree form, gives two identities per variable X at a fixed point:

$$\mathrm{BEL}(X) = \Pr'(X \mid \mathbf{e}'), \qquad \mathrm{BEL}(X\mathbf{U}) = \Pr'(X\hat{\mathbf{U}} \mid \mathbf{e}'),$$

the second replacing each parent by its clone. At $X = B$ the second identity is the table of (b), whose B-sum is $\Pr'(\hat{A} \mid \mathbf{e}')$; that sum is $\mathrm{BEL}(A)$ by the consistency constraints of Theorem 14.2; and the first identity at $X = A$ gives $\mathrm{BEL}(A) = \Pr'(A \mid \mathbf{e}')$. Chaining,

$$\Pr'(\hat{A} \mid \mathbf{e}') = \sum_B \mathrm{BEL}(AB) = \mathrm{BEL}(A) = \Pr'(A \mid \mathbf{e}'),$$

the node and its clone carrying the same posterior – the first equation of Exercise 14.12(b). Exact inference in the polytree confirms it: .71357 against .71349, the discrepancy being the four-digit rounding of Figure 14.8.
(d) The pair $\{A, \hat{A}\}$. Exact inference in the polytree $\mathcal{N}'$ gives $\Pr'(A\hat{A} \mid \mathbf{e}')$ directly, which is what Section 14.8.4 needs to score the deleted edge by

$$\mathrm{MI}(A; \hat{A} \mid \mathbf{e}') = \sum_{a\hat{a}} \Pr'(a\hat{a} \mid \mathbf{e}') \log \frac{\Pr'(a\hat{a} \mid \mathbf{e}')}{\Pr'(a \mid \mathbf{e}') \Pr'(\hat{a} \mid \mathbf{e}')}.$$

IBP in $\mathcal{N}$ produces only node and family marginals, and $\hat{A}$ is not a variable of $\mathcal{N}$ – the two ends of $A \to B$ are a single node there.

Problem 14.11. Let $\mathcal{N}'$ be a network that results from deleting a single edge $U \to X$ from the network $\mathcal{N}$ and suppose that this edge splits the network $\mathcal{N}$ into two disconnected subnetworks. As in Section 14.8.1, deletion removes the edge, adds a clone $\hat{U}$ (a root with CPT $\Theta_{\hat{U}}$, taking the same values as U) as the new parent of X, and adds an observed binary node $\hat{S}$ with CPT $\Theta_{\hat{S}|U}$ as a child of U. Show that $\Pr(\mathbf{e}_X \mid u) = \Pr'(\mathbf{e}_X \mid \hat{u})$, where $\mathbf{e}_X$ is evidence that instantiates nodes on the X-side of edge $U \to X$ and u and $\hat{u}$ denote the same value of U and of its clone.

Solution. Both sides equal $L(u) = \sum_{\mathbf{x}_X \sim \mathbf{e}_X} \Phi_X(\mathbf{x}_X, u)$, the likelihood the X-side subnetwork assigns to $\mathbf{e}_X$ as a function of the value crossing the deleted edge. Write $\mathbf{X}_U$, $\mathbf{X}_X$ for the nodes of the two subnetworks. Since $U \to X$ is the only edge across the split, every node of $\mathbf{X}_U$ has all its parents in $\mathbf{X}_U$, and every node of $\mathbf{X}_X$ has all its parents in $\mathbf{X}_X$ except X, whose parents are $\mathbf{P} \cup \{U\}$ with $\mathbf{P} \subseteq \mathbf{X}_X$. So the chain rule for $\mathcal{N}$ splits as

$$\Pr(\mathbf{x}_U, \mathbf{x}_X) = \Phi_U(\mathbf{x}_U) \cdot \Phi_X(\mathbf{x}_X, u),$$

where

$$\Phi_U(\mathbf{x}_U) = \prod_{V \in \mathbf{X}_U} \theta_{v|\mathbf{u}_V}, \qquad \Phi_X(\mathbf{x}_X, u) = \prod_{V \in \mathbf{X}_X} \theta_{v|\mathbf{u}_V},$$

with u the value $\mathbf{x}_U$ assigns to U, entering Φ_X only through X's CPT. Both pieces normalize – Φ_U is the joint of $\mathcal{N}_U$ and $\Phi_X(\cdot, u)$ that of $\mathcal{N}_X$ with $U = u$ fixed – so summing the joint of $\mathcal{N}$ over instantiations consistent with u and $\mathbf{e}_X$,

$$\begin{aligned}\Pr(\mathbf{e}_X, u) &= \sum_{\mathbf{x}_U \sim u} \sum_{\mathbf{x}_X \sim \mathbf{e}_X} \Phi_U(\mathbf{x}_U)\Phi_X(\mathbf{x}_X, u) \\ &= \Big(\sum_{\mathbf{x}_U \sim u} \Phi_U(\mathbf{x}_U) \Big) \cdot L(u) = \Pr(u)\, L(u),\end{aligned}$$

the factorization being legitimate because $\Phi_X(\cdot, u)$ does not depend on $\mathbf{x}_U$ once u is fixed, and $\sum_{\mathbf{x}_U \sim u} \Phi_U = \Pr(u)$ because summing Φ_X out of the joint gives 1. Hence $\Pr(\mathbf{e}_X \mid u) = L(u)$. In $\mathcal{N}'$ the components are $\mathbf{X}_U \cup \{\hat{S}\}$ and $\mathbf{X}_X \cup \{\hat{U}\}$, and the chain rule reads

$$\Pr'(\mathbf{x}_U, \hat{s}^*, \hat{u}, \mathbf{x}_X) = \Phi_U(\mathbf{x}_U)\, \theta_{\hat{s}^*|u} \cdot \theta_{\hat{u}}\, \Phi_X(\mathbf{x}_X, \hat{u}),$$

since every CPT of $\mathcal{N}$ carries over unchanged, X's now indexed by $\hat{u}$ – the same Φ_X, evaluated at $\hat{u}$. As $\mathbf{e}_X$ mentions no variable outside $\mathbf{X}_X$, summing out $\mathbf{X}_U$ and $\hat{S}$, which contribute $\sum_{\mathbf{x}_U} \Phi_U \sum_{\hat{s}^*} \theta_{\hat{s}^*|u} = 1$, leaves $\Pr'(\mathbf{e}_X, \hat{u}) = \theta_{\hat{u}} L(\hat{u})$; and $\hat{U}$ is a root, so $\Pr'(\hat{u}) = \theta_{\hat{u}}$ and $\Pr'(\mathbf{e}_X \mid \hat{u}) = L(\hat{u})$. Both sides therefore equal $L(u)$ at $u = \hat{u}$, for every choice of edge parameters – neither appears in L. □

Problem 14.12. Consider Equations 14.8 and 14.9, which govern the parameters of a deleted edge $U \to X$ in the approximate network $\mathcal{N}'$ (recall that $\hat{U}$ is a root of $\mathcal{N}'$ and the new parent of X, and $\hat{S}$ is a leaf of $\mathcal{N}'$ whose single parent is U and which is observed at value $\hat{s}$ in $\mathbf{e}'$):

$$\theta_{\hat{U}} = \Pr'(U \mid \mathbf{e}' - \hat{S}), \qquad \theta_{\hat{s}|U} = \eta \Pr'(\mathbf{e}' \mid \hat{U}).$$

(a) Show that Equations 14.8 and 14.9 are equivalent to

$$\theta_{\hat{u}} = \frac{\partial \Pr'(\mathbf{e}')}{\partial \theta_{\hat{s}|u}}, \qquad \theta_{\hat{s}|u} = \frac{\partial \Pr'(\mathbf{e}')}{\partial \theta_{\hat{u}}},$$

where $u = \hat{u}$.

(b) Show that Equations 14.8 and 14.9 are equivalent to

$$\begin{aligned}\Pr'(U \mid \mathbf{e}') &= \Pr'(\hat{U} \mid \mathbf{e}') \\ \Pr'(U \mid \mathbf{e}' - \hat{S}) &= \Pr'(\hat{U}).\end{aligned}$$

Solution. Both characterizations follow from two structural facts (throughout $\theta_{\hat{u}} > 0$ and $\Pr'(\mathbf{e}' - \hat{S}, u) > 0$).
Fact 1. $\hat{S}$ is a leaf with sole parent U, and $\hat{s} \in \mathbf{e}'$, so

$$\Pr'(\mathbf{e}', u) = \Pr'(\mathbf{e}' - \hat{S}, \hat{s}, u) = \theta_{\hat{s}|u} \Pr'(\mathbf{e}' - \hat{S}, u),$$

($\hat{S}$ is independent of the rest given U, and retracting it leaves it barren, so $\Pr'(\mathbf{e}' - \hat{S}, u)$ does not depend on $\Theta_{\hat{S}|U}$).
Fact 2. $\hat{U}$ is a root, so $\Pr'(\hat{u}) = \theta_{\hat{u}}$ and $\Pr'(\mathbf{e}', \hat{u}) = \theta_{\hat{u}} \Pr'(\mathbf{e}' \mid \hat{u})$.
(a) By Theorem 12.2 in the form of Equation 12.5, for $\theta_{x|\mathbf{u}} \neq 0$,

$$\frac{\partial \Pr(\mathbf{e})}{\partial \theta_{x|\mathbf{u}}} = \frac{\Pr(x, \mathbf{u}, \mathbf{e})}{\theta_{x|\mathbf{u}}}.$$

For $\theta_{\hat{s}|u}$, whose family is $\hat{S}U$,

$$\frac{\partial \Pr'(\mathbf{e}')}{\partial \theta_{\hat{s}|u}} = \frac{\Pr'(\hat{s}, u, \mathbf{e}')}{\theta_{\hat{s}|u}} = \frac{\Pr'(\mathbf{e}', u)}{\theta_{\hat{s}|u}} = \Pr'(\mathbf{e}' - \hat{S}, u),$$

the middle step by $\hat{s} \in \mathbf{e}'$ and the last by Fact 1; splitting $\Pr'(\mathbf{e}' - \hat{S}, u) = \Pr'(\mathbf{e}' - \hat{S}) \Pr'(u \mid \mathbf{e}' - \hat{S})$,

$$\frac{\partial \Pr'(\mathbf{e}')}{\partial \theta_{\hat{s}|u}} = \Pr'(\mathbf{e}' - \hat{S}) \cdot \Pr'(u \mid \mathbf{e}' - \hat{S}). \quad (*)$$

For $\theta_{\hat{u}}$, whose family is the singleton $\hat{U}$,

$$\frac{\partial \Pr'(\mathbf{e}')}{\partial \theta_{\hat{u}}} = \frac{\Pr'(\hat{u}, \mathbf{e}')}{\theta_{\hat{u}}} = \Pr'(\mathbf{e}' \mid \hat{u}), \quad (**)$$

by Fact 2. Now $(**)$ is Equation 14.9 with $\eta = 1$, and η is immaterial for conditional queries by Exercise 14.9; while $(*)$ makes the derivative proportional in u to $\Pr'(U \mid \mathbf{e}' - \hat{S})$ with u-independent constant $\Pr'(\mathbf{e}' - \hat{S})$, so normalization of $\Theta_{\hat{U}}$ forces Equation 14.8, and conversely 14.8 makes $(*)$ equal $\Pr'(\mathbf{e}' - \hat{S})\theta_{\hat{u}}$. Both derivative equations are thus read up to positive rescaling – harmlessly, since rescaling $\Theta_{\hat{U}}$ by $c > 0$ multiplies every chain-rule term of $\mathcal{N}'$ by c (each instantiation uses exactly one $\theta_{\hat{u}}$), and rescaling $\Theta_{\hat{S}|U}$ is covered by Exercise 14.9.
(b) By Fact 2, $\Pr'(\hat{U}) = \theta_{\hat{U}}$, so the second equation of (b) is Equation 14.8 verbatim (read at $u = \hat{u}$, as always). For the first, Fact 1 gives

$$\Pr'(u \mid \mathbf{e}') = \frac{\Pr'(\mathbf{e}', u)}{\Pr'(\mathbf{e}')} = \frac{\theta_{\hat{s}|u} \Pr'(\mathbf{e}' - \hat{S}, u)}{\Pr'(\mathbf{e}')},$$

and Fact 2 gives $\Pr'(\hat{u} \mid \mathbf{e}') = \theta_{\hat{u}} \Pr'(\mathbf{e}' \mid \hat{u}) / \Pr'(\mathbf{e}')$, so, cancelling $\Pr'(\mathbf{e}')$, the first equation holds iff for every $u = \hat{u}$,

$$\theta_{\hat{s}|u} \Pr'(\mathbf{e}' - \hat{S}, u) = \theta_{\hat{u}} \Pr'(\mathbf{e}' \mid \hat{u}). \quad (\dagger)$$

Assume Equation 14.8, i.e. $\theta_{\hat{u}} = \Pr'(\mathbf{e}' - \hat{S}, u) / \Pr'(\mathbf{e}' - \hat{S})$; substituting into the right of $(\dagger)$ and cancelling the positive $\Pr'(\mathbf{e}' - \hat{S}, u)$ turns $(\dagger)$ into

$$\theta_{\hat{s}|u} = \frac{1}{\Pr'(\mathbf{e}' - \hat{S})} \Pr'(\mathbf{e}' \mid \hat{u}),$$

Equation 14.9 with $\eta = 1/A$, $A = \Pr'(\mathbf{e}' - \hat{S}) > 0$. Conversely η is not free once 14.8 is imposed: under 14.8 and 14.9 with some $\eta > 0$ the two sides of $(\dagger)$ are $\eta \Pr'(\mathbf{e}' \mid \hat{u}) \Pr'(\mathbf{e}' - \hat{S}, u)$ and $A^{-1} \Pr'(\mathbf{e}' - \hat{S}, u) \Pr'(\mathbf{e}' \mid \hat{u})$, a ratio of ηA at every u; but summing either side over u gives $\Pr'(\mathbf{e}')$ in both cases (by Fact 1 on the left, Fact 2 on the right), so $\eta A = 1$ and $(\dagger)$ holds term by term. Hence 14.8 and 14.9 give both equations of (b), and conversely the second gives 14.8 and then $(\dagger)$ gives 14.9. □

Problem 14.13. Let $\mathcal{N}$ be a Bayesian network and $\mathcal{N}'$ be the result of deleting a set Δ of edges in $\mathcal{N}$ (each deleted edge $U \to X$ contributing a root clone $\hat{U}$ as the new parent of X, with CPT $\Theta_{\hat{U}}$, and an observed leaf $\hat{S}$ with parent U and CPT $\Theta_{\hat{S}|U}$; the evidence is $\mathbf{e}'$, which is $\mathbf{e}$ extended by $\hat{s}$ for every added node $\hat{S}$). Show how an ED-BP fixed point in $\mathcal{N}'$ corresponds to a fixed point of IJGP. In particular, show how to construct a joingraph that allows IJGP, Algorithm 41, to simulate Algorithm 42, ED-BP.

Solution. Take a jointree for $\mathcal{N}'$, relabel away the auxiliary nodes, and restore each deleted edge as one extra joingraph edge whose two messages are its two edge parameters; IJGP's exclusion rule (a message out of an edge is computed without the message that came in on it) is then exactly the retraction $\mathbf{e}' - \hat{S}$ of Equation 14.8 and the division by $\theta_{\hat{u}}$ of Equation 14.9.

Construction. Let T' be a jointree for $\mathcal{N}'$. By the jointree property every family of $\mathcal{N}'$ sits in some cluster, so for each deleted edge $U \to X$ there are:

- a cluster C_i containing the family $\{U, \hat{S}\}$, to which $\Theta_{\hat{S}|U}$ and the indicator $\lambda_{\hat{s}}$ are assigned;
- a cluster C_j containing the family $\{X\} \cup \hat{\mathbf{U}}$ of X in $\mathcal{N}'$, hence containing $\hat{U}$, to which $\Theta_{\hat{U}}$ is assigned.

Build G as follows. (1) One node per cluster of T'; delete every $\hat{S}$ from every cluster and separator and rename every clone $\hat{U}$ back to U, keeping the edges of T' with their relabelled separators. (2) For each deleted edge $U \to X$, add an edge between the relabelled i and j above, with separator $S_{ij} = \{U\}$. (3) Assign each CPT and indicator of $\mathcal{N}$ to the cluster that carried the corresponding factor of $\mathcal{N}'$ (same table, reindexed by the renaming); discard $\Theta_{\hat{U}}$, $\Theta_{\hat{S}|U}$ and $\lambda_{\hat{s}}$.

G satisfies Definition 14.1. Property 1 holds by construction. Property 2: the family $\{V\} \cup \hat{\mathbf{U}}$ of V in $\mathcal{N}'$ lies in some cluster of T', and renaming clones back turns it into V's family in $\mathcal{N}$. Property 4: retained separators still sit inside the relabelled cluster intersections, and a new edge's $\{V\}$ lies in both its endpoints.

Property 3 is the substantive one. If no edge out of V was deleted, V's clusters form a connected subtree of T' and relabelling does not disturb it. If $V \to X_1, \ldots, V \to X_m$ were deleted, the occurrences of V form a subtree T_V and those of each clone $\hat{V}_k$ a subtree $T_{\hat{V}_k}$, so after renaming V occupies the disconnected $T_V \cup T_{\hat{V}_1} \cup \cdots \cup T_{\hat{V}_m}$. But step 2 adds, for each k, an edge of separator $\{V\}$ from $i_k \in T_V$ (home of the family $\{V, \hat{S}_k\}$, hence containing V) to $j_k \in T_{\hat{V}_k}$ (containing $\hat{V}_k$), attaching every clone subtree to T_V. Two clusters containing V are then joined either by a jointree path inside one subtree, or by walking to j_k, crossing the new edge to i_k, and walking inside T_V (concatenating two such walks for two distinct clone subtrees); every cluster and separator on such a path contains V. (Check!)

IJGP on G simulates ED-BP on $\mathcal{N}'$. Fix edge parameters $\Theta_{\hat{U}}, \Theta_{\hat{S}|U}$ for every deleted edge, let $\Pr'$ be the distribution they induce, and define candidate IJGP messages:

- On a new edge $i - j$ created for the deleted edge $U \to X$: $M_{ij} := \Theta_{\hat{U}}$ (the message from the U-side to the X-side) and $M_{ji} := \Theta_{\hat{s}|U}$ (the message from the X-side back to the U-side). Both are functions of the single separator variable U, as required.
- On a retained T'-edge: the corresponding jointree message computed by exact inference in $\mathcal{N}'$ with these parameters.

These are consistent: the only difference between C_i in T' and in G is that G strips the factor $\Theta_{\hat{S}|U}\lambda_{\hat{s}}$ – a function of U alone, of value $\theta_{\hat{s}|U}$ – and supplies exactly that function as the incoming M_{ji}, and likewise at C_j the factor $\Theta_{\hat{U}}$ is replaced by the incoming M_{ij}. Both algorithms multiply the cluster factor by the incoming messages before summing out, so by induction along T' every retained-edge IJGP message equals the corresponding jointree message. From i to j the IJGP update excludes $M_{ji} = \theta_{\hat{s}|U}$:

$$M_{ij} \;=\; \eta \sum_{C_i \setminus \{U\}} \Phi_i \prod_{k \neq j} M_{ki}.$$

The right-hand side is the jointree computation of $\Pr'(U, \mathbf{e}' - \hat{S})$ at C_i – it uses every factor and message of $\mathcal{N}'$ except the CPT and indicator of $\hat{S}$, and dropping those is exactly retraction, $\hat{S}$ being then a barren leaf – so after normalization $M_{ij} = \Pr'(U \mid \mathbf{e}' - \hat{S})$, the ED-BP update for $\Theta_{\hat{U}}$. From j to i the

update excludes $M_{ij} = \theta_{\hat{u}}$:

$$M_{ji} \;=\; \eta \sum_{C_j \setminus \{U\}} \Phi_j \prod_{k \neq i} M_{kj}.$$

Its right-hand side is the jointree computation of $\Pr'(\hat{U}, \mathbf{e}')$ at C_j with $\Theta_{\hat{U}}$ omitted, i.e. $\Pr'(\hat{U}, \mathbf{e}')/\theta_{\hat{U}} = \Pr'(\mathbf{e}' \mid \hat{U})$ – legitimate since $\hat{U}$ is a root, so $\Theta_{\hat{U}}$ is a function of the separator variable alone and factors out of the summation. Hence $M_{ji} = \eta \Pr'(\mathbf{e}' \mid \hat{U})$, Equation 14.9.
Fixed points therefore correspond. If the edge parameters satisfy 14.8 and 14.9, the messages above reproduce themselves – retained-edge ones as jointree messages of a converged exact computation, new-edge ones by the two displays – so they are an IJGP fixed point on G; conversely, setting $\Theta_{\hat{U}} := M_{ij}$ and $\Theta_{\hat{s}|U} := M_{ji}$ at an IJGP fixed point makes the retained-edge messages the jointree messages of $\mathcal{N}'$ under those parameters, and the same two displays turn the new-edge equations into 14.8 and 14.9. The joingraph beliefs are then the exact approximate-network marginals,

$$\mathrm{BEL}(C_i) = \Pr'(C_i \mid \mathbf{e}'), \qquad \mathrm{BEL}(S_{ij}) = \eta\, \theta_{\hat{U}} \theta_{\hat{s}|U} = \Pr'(U \mid \mathbf{e}')$$

on a new edge, each variable of G read as the $\mathcal{N}'$ variable it was renamed from, so that a cluster C_j carrying the family of X reports $\Pr'(X\hat{\mathbf{U}} \mid \mathbf{e}')$. The second equality is the separator form of Exercise 14.12(b): $\theta_{\hat{u}}\theta_{\hat{s}|u} \propto \Pr'(u \mid \mathbf{e}' - \hat{S}) \Pr'(\mathbf{e}' \mid \hat{u}) \propto \Pr'(\mathbf{e}' - \hat{S}, u)\theta_{\hat{s}|u} = \Pr'(\mathbf{e}', u)$ by Fact 1 of that exercise. (Per-iteration correspondence additionally requires a schedule completing a full inward-outward sweep of the tree part of G between successive new-edge updates; Algorithm 41's parallel schedule does not, though any sequential one with that property does.)

Problem 14.14. Let $\mathcal{N}'$ be a Bayesian network that results from deleting an edge $U \to X$ that splits network $\mathcal{N}$ into two disconnected subnetworks: $\mathcal{N}_U$ containing U and $\mathcal{N}_X$ containing X. Let $\mathbf{X}_U$ be the variables of subnetwork $\mathcal{N}_U$ and $\mathbf{X}_X$ be the variables of subnetwork $\mathcal{N}_X$. As usual, deletion adds a root clone $\hat{U}$ with CPT $\Theta_{\hat{U}}$ as the new parent of X, and an observed leaf $\hat{S}$ with CPT $\Theta_{\hat{S}|U}$ as a child of U; the evidence $\mathbf{e}$ of $\mathcal{N}$ splits as $\mathbf{e} = \mathbf{e}_U, \mathbf{e}_X$ over the two sides, and $\mathbf{e}' = \mathbf{e}, \hat{s}$.
If the parameters $\Theta_{\hat{U}}$ and $\Theta_{\hat{S}|U}$ are determined by Equations 14.8 and 14.9, show that the marginal distributions for each approximate subnetwork are exact:

$$\Pr'(\mathbf{X}_U \mid \mathbf{e}') = \Pr(\mathbf{X}_U \mid \mathbf{e})$$
$$\Pr'(\mathbf{X}_X \mid \mathbf{e}') = \Pr(\mathbf{X}_X \mid \mathbf{e}).$$

Solution. Each side's joint is a constant multiple of the corresponding joint in $\mathcal{N}$; the constant cancels on normalizing. As in Exercise 14.11, the deleted edge is the only edge of $\mathcal{N}$ crossing between $\mathbf{X}_U$ and $\mathbf{X}_X$, so the chain rule splits as

$$\Pr(\mathbf{x}_U, \mathbf{x}_X) = \Phi_U(\mathbf{x}_U)\, \Phi_X(\mathbf{x}_X, u),$$

with Φ_U the product of the CPTs of $\mathcal{N}_U$, Φ_X the product of the CPTs of $\mathcal{N}_X$ (the value u entering only through X's own CPT), and $\sum_{\mathbf{x}_U} \Phi_U = 1$, $\sum_{\mathbf{x}_X} \Phi_X(\cdot, u) = 1$ for each u. In $\mathcal{N}'$ the same two factors reappear, with $\hat{u}$ in place of u:

$$\Pr'(\mathbf{x}_U, \hat{s}, \hat{u}, \mathbf{x}_X) = \Phi_U(\mathbf{x}_U)\, \theta_{\hat{s}|u} \cdot \theta_{\hat{u}}\, \Phi_X(\mathbf{x}_X, \hat{u}),$$

where u is the value $\mathbf{x}_U$ assigns to U. With $L(u) = \sum_{\mathbf{x}_X \sim \mathbf{e}_X} \Phi_X(\mathbf{x}_X, u)$, Exercise 14.11 gives $L(u) = \Pr(\mathbf{e}_X \mid u) = \Pr'(\mathbf{e}_X \mid \hat{u})$.
The edge parameters. The components of $\mathcal{N}'$ are $\mathbf{X}_U \cup \{\hat{S}\}$ and $\mathbf{X}_X \cup \{\hat{U}\}$, so splitting $\mathbf{e}' = \mathbf{e}_U, \hat{s}, \mathbf{e}_X$ across them gives $\Pr'(\mathbf{e}' \mid \hat{u}) = \Pr'(\mathbf{e}_U, \hat{s})L(\hat{u})$ with the first factor independent of $\hat{u}$; so Equation 14.9 says

$$\theta_{\hat{s}|u} = \eta_1 L(u) \quad \text{for a constant } \eta_1 > 0. \quad (1)$$

For Equation 14.8, $\mathbf{e}' - \hat{S} = \mathbf{e}_U, \mathbf{e}_X$ and U lies in the component carrying $\mathbf{e}_U$, so $\Pr'(U \mid \mathbf{e}_U, \mathbf{e}_X) = \Pr'(U \mid \mathbf{e}_U)$; and retracting $\hat{S}$ makes it barren, so within its component the distribution is that of $\mathcal{N}_U$,

giving $\Pr'(u, \mathbf{e}_U) = \sum_{\mathbf{x}_U \sim u, \mathbf{e}_U} \Phi_U = \Pr(u, \mathbf{e}_U)$. Hence

$$\theta_{\hat{u}} = \Pr(u \mid \mathbf{e}_U) = \frac{\Pr(u, \mathbf{e}_U)}{\Pr(\mathbf{e}_U)}. \quad (2)$$

The U-side. Let $\mathbf{x}_U$ be consistent with $\mathbf{e}_U$ (otherwise both sides vanish). Summing the $\mathcal{N}'$ joint over $\hat{U}$ and over the $\mathbf{x}_X$ consistent with $\mathbf{e}_X$,

$$\begin{aligned}\Pr'(\mathbf{x}_U, \mathbf{e}') &= \Phi_U(\mathbf{x}_U)\, \theta_{\hat{s}|u} \sum_{\hat{u}} \theta_{\hat{u}} \sum_{\mathbf{x}_X \sim \mathbf{e}_X} \Phi_X(\mathbf{x}_X, \hat{u}) \\ &= \Phi_U(\mathbf{x}_U)\, \theta_{\hat{s}|u} \cdot K, \qquad K := \sum_{\hat{u}} \theta_{\hat{u}} L(\hat{u}).\end{aligned}$$

Substituting (1) gives $\Pr'(\mathbf{x}_U, \mathbf{e}') = \eta_1 K\, \Phi_U(\mathbf{x}_U) L(u)$, whereas in $\mathcal{N}$, $\Pr(\mathbf{x}_U, \mathbf{e}) = \Phi_U(\mathbf{x}_U) \sum_{\mathbf{x}_X \sim \mathbf{e}_X} \Phi_X(\mathbf{x}_X, u) = \Phi_U(\mathbf{x}_U) L(u)$. So $\Pr'(\mathbf{x}_U, \mathbf{e}') = \eta_1 K \Pr(\mathbf{x}_U, \mathbf{e})$ with $\eta_1 K > 0$ independent of $\mathbf{x}_U$; summing over $\mathbf{x}_U$ gives $\Pr'(\mathbf{e}') = \eta_1 K \Pr(\mathbf{e})$, and dividing, $\Pr'(\mathbf{x}_U \mid \mathbf{e}') = \Pr(\mathbf{x}_U \mid \mathbf{e})$ – the first claim, using only Equation 14.9.

The X-side. Let $\mathbf{x}_X$ be consistent with $\mathbf{e}_X$. Summing the $\mathcal{N}'$ joint over $\hat{U}$ and over the $\mathbf{x}_U$ consistent with $\mathbf{e}_U$, the two components factor:

$$\begin{aligned}\Pr'(\mathbf{x}_X, \mathbf{e}') &= \Big(\sum_{\mathbf{x}_U \sim \mathbf{e}_U} \Phi_U(\mathbf{x}_U) \theta_{\hat{s}|u} \Big) \cdot \sum_{\hat{u}} \theta_{\hat{u}} \Phi_X(\mathbf{x}_X, \hat{u}) \\ &= C \sum_{u} \theta_{\hat{u}} \Phi_X(\mathbf{x}_X, u),\end{aligned}$$

where $C = \Pr'(\mathbf{e}_U, \hat{s}) > 0$ does not depend on $\mathbf{x}_X$. Substituting (2) turns this into $\big(C / \Pr(\mathbf{e}_U)\big) \sum_u \Pr(u, \mathbf{e}_U) \Phi_X(\mathbf{x}_X, u)$, whereas in $\mathcal{N}$, grouping the $\mathbf{x}_U$ by the value u they assign to U,

$$\Pr(\mathbf{x}_X, \mathbf{e}) = \sum_{\mathbf{x}_U \sim \mathbf{e}_U} \Phi_U(\mathbf{x}_U) \Phi_X(\mathbf{x}_X, u) = \sum_u \Pr(u, \mathbf{e}_U)\, \Phi_X(\mathbf{x}_X, u).$$

Hence $\Pr'(\mathbf{x}_X, \mathbf{e}') = \big(C / \Pr(\mathbf{e}_U)\big) \Pr(\mathbf{x}_X, \mathbf{e})$, again with a constant independent of $\mathbf{x}_X$; summing over $\mathbf{x}_X$ and dividing gives $\Pr'(\mathbf{x}_X \mid \mathbf{e}') = \Pr(\mathbf{x}_X \mid \mathbf{e})$, the second claim, using only Equation 14.8. □

13.3 Exercises 14.15–14.20

Problem 14.15. Recall the edge-deletion construction of Section 14.8: deleting an edge $U \to X$ from network N removes the edge and adds two variables — a clone $\hat{U}$ that has the same values as U and becomes a parent of X (a root node with CPT $\theta_{\hat{u}}$), and a binary variable $\hat{S}$ that is a child of U (with CPT $\theta_{\hat{s}|u}$) and is observed to $\hat{s}$. The extended evidence is e', consisting of the original evidence e together with the value $\hat{s}$ for each added variable $\hat{S}$.

Let N′ be a polytree that results from deleting edges in network N, and suppose that the parameter edges of N′ satisfy Equations 14.8 and 14.9,

$$\Theta_{\hat{U}} \;=\; Pr'(U \mid e' - \hat{S}), \qquad \Theta_{\hat{s}|U} \;=\; \eta\, Pr'(e' \mid \hat{U}) \;\text{ for some } \eta > 0.$$

Suppose further that we are using the polytree algorithm (i.e., belief propagation) to compute the MI scores in Algorithm 43, where a deleted edge $U \to X$ is ranked by

$$\mathrm{MI}(U; \hat{U} \mid e') \;=\; \sum_{u\hat{u}} Pr'(u\hat{u} \mid e') \log \frac{Pr'(u\hat{u} \mid e')}{Pr'(u \mid e')\, Pr'(\hat{u} \mid e')}.$$

To compute the MI score for each edge $U \to X$ that we delete, we need two types of values:

- Node marginals $Pr'(u \mid e')$ and $Pr'(\hat{u} \mid e')$.
- Joint marginals $Pr'(u\hat{u} \mid e')$.

If network N has n nodes and m edges, and since N′ is a polytree, at most $n-1$ edges are left undeleted. Thus, we may need to score $O(n^2)$ deleted edges in the worst case. Luckily, to compute every node marginal we only need to run belief propagation once. However, if we naively computed joint marginals $Pr'(u\hat{u} \mid e')$, we could have run belief propagation $O(m \max_U |U|^2)$ times: for each of our $O(m)$ deleted edges, we could have run belief propagation once for each of the $|U|^2$ instantiations $u\hat{u}$. Consider the following:

$$Pr'(u\hat{u} \mid e') \;=\; Pr'(\hat{u} \mid u, e')\, Pr'(u \mid e').$$

Show that we can compute $Pr'(\hat{u} \mid u, e')$ for all instantiations $u\hat{u}$ using only $O(n \max_U |U|)$ runs of belief propagation, thus showing that we need only the same number of runs of belief propagation to score all edges in Algorithm 43.

Solution. Index the runs by tail variables, not by deleted edges or by pairs: one run of belief propagation on the polytree N′ returns the exact marginal of **every** node simultaneously, clones included, for whatever evidence it is conditioned on. Let $\mathcal{T}$ be the set of tails of deleted edges, so $|\mathcal{T}| \leq n$, every tail being a node of N, and run:

- Run 0: evidence e'.
- For each $U \in \mathcal{T}$ and each state u with $Pr'(u \mid e') > 0$: one run with evidence $e' \cup \{U = u\}$.

That is $1 + \sum_{U \in \mathcal{T}} |U| \leq 1 + n \max_U |U| = O(n \max_U |U|)$ runs. Run 0 delivers $Pr'(u \mid e')$ and $Pr'(\hat{u} \mid e')$ for every tail and every clone at once. Adding evidence leaves the structure a polytree, so belief propagation under $e' \cup \{U = u\}$ is still exact, and that single run reports $Pr'(\hat{u} \mid u, e')$ for every clone $\hat{U}$ of U and every $\hat{u}$ simultaneously – so the $|U|$ runs indexed by the states of U serve all deleted edges with tail U, replacing the $|U|^2$ runs indexed by pairs. States with $Pr'(u \mid e') = 0$ may be skipped: they make $Pr'(u\hat{u} \mid e') = 0$, which contributes nothing to MI under $0 \log 0 = 0$.
The chain rule $Pr'(u\hat{u} \mid e') = Pr'(\hat{u} \mid u, e')Pr'(u \mid e')$ then assembles every score from stored numbers, at $O(|U|^2)$ arithmetic per edge and no further inference – $O(n^2 \max_U |U|^2)$ arithmetic overall, but only $O(n \max_U |U|)$ runs of belief propagation, as claimed.

Problem 14.16. Let N and N′ be two Bayesian networks that have the same structure but possibly different CPTs. Let Pr and Pr' be their induced distributions, and recall that the KL divergence between two distributions over the same set of variables $\mathbf{X}$ is

$$KL(Pr, Pr') \;=\; \sum_{\mathbf{x}} Pr(\mathbf{x}) \log \frac{Pr(\mathbf{x})}{Pr'(\mathbf{x})}.$$

Prove that

$$\begin{aligned} KL(Pr, Pr') \;&=\; \sum_{X\mathbf{U}} \sum_{\mathbf{u}} Pr(\mathbf{u}) \cdot KL\Big(\Theta_{X|\mathbf{u}},\, \Theta'_{X|\mathbf{u}}\Big) \\ &=\; \sum_{X\mathbf{U}} \sum_{\mathbf{u}} Pr(\mathbf{u}) \cdot \sum_{x} \theta_{x|\mathbf{u}} \log \frac{\theta_{x|\mathbf{u}}}{\theta'_{x|\mathbf{u}}}, \end{aligned}$$

where $\Theta_{X|\mathbf{U}}$ are CPTs in network N, $\Theta'_{X|\mathbf{U}}$ are the corresponding CPTs in network N′, and $X\mathbf{U}$ ranges over the families of the (shared) network structure.

Solution. Substitute the chain rule into the definition of KL and regroup by family. The shared structure gives the same families in both networks, so by Equation 4.2 $Pr(\mathbf{x}) = \prod_{X\mathbf{U}} \theta_{x|\mathbf{u}}$ and $Pr'(\mathbf{x}) = \prod_{X\mathbf{U}} \theta'_{x|\mathbf{u}}$ with $x\mathbf{u} \sim \mathbf{x}$; if $\theta'_{x|\mathbf{u}} = 0 < \theta_{x|\mathbf{u}}$ for some family both sides are $+\infty$, so assume not, and use

$0 \log 0 = 0$. Taking logarithms, $\log(Pr(\mathbf{x})/Pr'(\mathbf{x})) = \sum_{X\mathbf{U}} \log(\theta_{x|\mathbf{u}}/\theta'_{x|\mathbf{u}})$, whence

$$\begin{aligned} KL(Pr, Pr') &= \sum_{X\mathbf{U}} \sum_{\mathbf{x}} Pr(\mathbf{x}) \log \frac{\theta_{x|\mathbf{u}}}{\theta'_{x|\mathbf{u}}} \\ &= \sum_{X\mathbf{U}} \sum_{x\mathbf{u}} Pr(x\mathbf{u}) \log \frac{\theta_{x|\mathbf{u}}}{\theta'_{x|\mathbf{u}}} \\ &= \sum_{X\mathbf{U}} \sum_{\mathbf{u}} Pr(\mathbf{u}) \sum_{x} \theta_{x|\mathbf{u}} \log \frac{\theta_{x|\mathbf{u}}}{\theta'_{x|\mathbf{u}}}, \end{aligned}$$

the second line grouping complete instantiations by the value they assign $X\mathbf{U}$ (the summand depending on $\mathbf{x}$ only through $x\mathbf{u}$), the third using $Pr(x\mathbf{u}) = Pr(\mathbf{u})\theta_{x|\mathbf{u}}$ – the Bayesian network semantics of Chapter 4, and trivially true when $Pr(\mathbf{u}) = 0$. Each CPT column being a distribution over the values of X, the inner sum is $KL(\Theta_{X|\mathbf{u}}, \Theta'_{X|\mathbf{u}})$, which is the first identity. □

Problem 14.17. Suppose we have a network N with a variable X that has parents Y and $\mathbf{U}$. Suppose that deleting the edge $Y \to X$ results in a network N′ where

- N′ has the same structure as N except that edge $Y \to X$ is removed.
- The CPT for variable X in N′ is given by $\theta'_{x|\mathbf{u}} \stackrel{\text{def}}{=} Pr(x \mid \mathbf{u})$.
- The CPTs for variables other than X in N′ are the same as those in N.

(a) Prove that when we delete a single edge $Y \to X$, we have

$$\begin{aligned} KL(Pr, Pr') &= \mathrm{MI}(X; Y \mid \mathbf{U}) \\ &= \sum_{xy\mathbf{u}} Pr(xy\mathbf{u}) \log \frac{Pr(xy \mid \mathbf{u})}{Pr(x \mid \mathbf{u}) Pr(y \mid \mathbf{u})}. \end{aligned}$$

(b) Prove that when we delete multiple edges but at most one edge $Y \to X$ incoming into X is deleted, the error is additive:

$$KL(Pr, Pr') = \sum_{Y \to X} \mathrm{MI}(X; Y \mid \mathbf{U}).$$

(c) Identify a small network conditioned on some evidence e where the KL divergence $KL(Pr(X|e), Pr'(X|e))$ can be made arbitrarily large even when the divergence $KL(Pr, Pr')$ can be made arbitrarily close to zero. Hint: a two-node network $A \to B$ suffices.

Solution. Exercise 14.16 needs a shared structure, so replace N′ by the network N″ with the structure of N, the CPT $\theta''_{x|y\mathbf{u}} = Pr(x \mid \mathbf{u})$ for X (the same value for every y, so every column sums to one) and the CPTs of N elsewhere: its chain-rule product is term by term that of N′, the X-factor merely carrying a vacuous extra argument y, so $Pr'' = Pr'$.
(a) Exercise 14.16 writes $KL(Pr, Pr') = KL(Pr, Pr'')$ as a sum over the families $Z\mathbf{V}$ of N, and every family other than $XY\mathbf{U}$ has $\theta''_{z|\mathbf{v}} = \theta_{z|\mathbf{v}}$, hence zero inner divergence. Only the family of X survives:

$$\begin{aligned} KL(Pr, Pr') &= \sum_{y\mathbf{u}} Pr(y\mathbf{u}) \sum_{x} \theta_{x|y\mathbf{u}} \log \frac{\theta_{x|y\mathbf{u}}}{\theta''_{x|y\mathbf{u}}} \\ &= \sum_{xy\mathbf{u}} Pr(y\mathbf{u})\, Pr(x \mid y\mathbf{u}) \log \frac{Pr(x \mid y\mathbf{u})}{Pr(x \mid \mathbf{u})} \\ &= \sum_{xy\mathbf{u}} Pr(xy\mathbf{u}) \log \frac{Pr(x \mid y\mathbf{u})}{Pr(x \mid \mathbf{u})}, \end{aligned}$$

using $\theta_{x|y\mathbf{u}} = Pr(x \mid y\mathbf{u})$ and $Pr(y\mathbf{u})Pr(x \mid y\mathbf{u}) = Pr(xy\mathbf{u})$. The chain rule inside the conditioning event $\mathbf{u}$ gives $Pr(x \mid y\mathbf{u}) = Pr(xy \mid \mathbf{u})/Pr(y \mid \mathbf{u})$, so the summand's ratio is $Pr(xy \mid$

$\mathbf{u})/[Pr(x \mid \mathbf{u})Pr(y \mid \mathbf{u})]$, which makes the sum $\mathrm{MI}(X;Y \mid \mathbf{U})$ by definition, since $\sum_{xy\mathbf{u}} Pr(xy\mathbf{u})f = \sum_{\mathbf{u}} Pr(\mathbf{u}) \sum_{xy} Pr(xy \mid \mathbf{u})f$.

(b) Let $\mathcal{D}$ be the deleted edges, no two sharing a head. For $Y \to X \in \mathcal{D}$ write $\mathbf{U}$ for X's other parents in N; by the stipulation these are exactly X's parents in N′, so $\theta'_{x|\mathbf{u}} = Pr(x \mid \mathbf{u})$ is well posed. Form N″ as before, putting $\theta''_{x|y\mathbf{u}} = Pr(x \mid \mathbf{u})$ at each head; again $Pr'' = Pr'$. Exercise 14.16 makes the divergence a sum over families in which a family contributes zero unless its CPT changed, and the changed families are exactly those of the heads – pairwise distinct, so no family is changed twice. The contribution of the family of head X is, by the computation of (a) verbatim,

$$\sum_{y\mathbf{u}} Pr(y\mathbf{u}) \sum_{x} \theta_{x|y\mathbf{u}} \log \frac{\theta_{x|y\mathbf{u}}}{Pr(x \mid \mathbf{u})} \;=\; \mathrm{MI}(X;Y \mid \mathbf{U}).$$

Summing over families,

$$KL(Pr, Pr') \;=\; \sum_{Y \to X \,\in\, \mathcal{D}} \mathrm{MI}(X;Y \mid \mathbf{U}),$$

which is the claimed additivity. The hypothesis is needed: two deleted edges $Y_1 \to X$, $Y_2 \to X$ into the same child would give the family of X one divergence term $\sum_{\mathbf{u}} Pr(\mathbf{u}) KL(\Theta_{X|y_1y_2\mathbf{u}}, \Theta'_{X|\mathbf{u}})$ that does not in general split into two mutual informations.

(c) Take $A \to B$ with A, B binary and $\delta \in (0, 1/2)$:

$$\theta_a = \delta, \qquad \theta_{\bar{a}} = 1 - \delta, \qquad \theta_{b|a} = 1, \qquad \theta_{b|\bar{a}} = \delta^2.$$

Delete $A \to B$, so N′ has two roots, A keeping its CPT and B getting $\theta'_b = Pr(b) = \delta + (1-\delta)\delta^2$; take $e = b$ and query A. In N′ the two variables are independent, so $Pr'(a \mid b) = Pr(a) = \delta$, while by Bayes' rule

$$Pr(a \mid b) = \frac{\delta}{\delta + (1-\delta)\delta^2} = \frac{1}{1 + (1-\delta)\delta} \;\longrightarrow\; 1 \quad (\delta \to 0).$$

With $p = Pr(a \mid b)$,

$$KL\big(Pr(A|b), Pr'(A|b)\big) = p \log \frac{p}{\delta} + (1-p) \log \frac{1-p}{1-\delta} \;\geq\; p \log \frac{p}{\delta} - \tfrac{1}{e},$$

using $t \log t \geq -1/e$ and $\log \frac{1}{1-\delta} > 0$; since $p \geq 1/2$ for small δ, the first term is at least $\frac{1}{2} \log \frac{1}{2\delta} \to +\infty$. Meanwhile by (a) with $\mathbf{U} = \emptyset$, $KL(Pr, Pr') = \mathrm{MI}(A;B)$, which is the average of the conditional divergences,

$$\mathrm{MI}(A;B) \;=\; Pr(b)\, KL(Pr(A|b), Pr(A)) \;+\; Pr(\bar{b})\, KL\big(Pr(A|\bar{b}), Pr(A)\big)\,.$$

The first term is $O(\delta \log(1/\delta))$, since $Pr(b) \leq 2\delta$ multiplies a divergence of order $\log(1/\delta)$; the second is $O(\delta)$, since $\theta_{\bar{b}|a} = 0$ forces $Pr(a \mid \bar{b}) = 0$, whence $KL(Pr(A|\bar{b}), Pr(A)) = \log \frac{1}{1-\delta}$. So $\mathrm{MI}(A;B) \to 0$. Numerically:

δ	$KL(Pr, Pr') = \mathrm{MI}(A;B)$	$KL(Pr(A \mid b), Pr'(A \mid b))$
10^{-2}	0.0554	4.505
10^{-3}	0.00790	6.893
10^{-4}	0.00102	9.208
10^{-6}	1.48×10^{-5}	13.815

(natural logarithms). Thus $KL(Pr, Pr')$ can be driven to zero while $KL(Pr(A|b), Pr'(A|b))$ grows without bound.

Problem 14.18. Recall the edge-deletion construction: deleting an edge $U \to X$ from network N removes the edge, adds a clone $\hat{U}$ (a root with the same values as U and CPT $\theta_{\hat{u}}$) as a parent of X, and adds a binary child $\hat{S}$ of U (with CPT $\theta_{\hat{s}|u}$) that is observed to $\hat{s}$. The evidence e' is the original evidence e plus $\hat{s}$ for each added $\hat{S}$.

Let N′ be a Bayesian network that results from deleting edges $U \to X$ from network N. Show that the edge parameters of network N′ satisfy Equations 14.8 and 14.9,

$$\Theta_{\hat{U}} = Pr'(U \mid e' - \hat{S}), \qquad \Theta_{\hat{s}|U} = \eta\, Pr'(e' \mid \hat{U}) \;\text{ for some constant } \eta > 0,$$

if and only if the edge parameters are a stationary point of $Pr'(e') \cdot \frac{1}{z}$, where z is given by Equation 14.16,

$$z \;=\; \prod_{U \to X} z_{UX}, \qquad z_{UX} \;=\; \sum_{u=\hat{u}} \theta_{\hat{u}}\, \theta_{\hat{s}|u},$$

the product ranging over all deleted edges.

Solution. Both conditions reduce to the single system (F) below, $\theta_{\hat{u}}\theta_{\hat{s}|u} = z_{UX} Pr'(u \mid e') = z_{UX} Pr'(\hat{u} \mid e')$. The free variables are the edge parameters $\theta_{\hat{u}}$ (one per state of each clone) and $\theta_{\hat{s}|u}$ (one per state of each tail), one group per deleted edge; all other CPTs are inherited and fixed, and we work at an interior point, every such parameter strictly positive and $Pr'(e') > 0$. Write $G = P/z$ with $P = Pr'(e')$, $z = \prod_{U \to X} z_{UX}$.
P is multilinear in the edge parameters. Expanding the network polynomial of N′ under evidence e',

$$P \;=\; \sum_{\mathbf{z}} \lambda_{e'}(\mathbf{z}) \prod_{V\mathbf{W}} \theta_{v|\mathbf{w}},$$

with $\lambda_{e'}(\mathbf{z}) \in \{0, 1\}$ recording compatibility, each surviving term carries exactly one factor $\theta_{\hat{u}}$ per clone and one $\theta_{\hat{s}|u}$ per $\hat{S}$ (terms with $\hat{S} = \bar{\hat{s}}$ being killed by $\lambda_{e'}$). Collecting the terms with $\hat{U} = \hat{u}$ and dividing by $\theta_{\hat{u}}$, and likewise for $\theta_{\hat{s}|u}$, gives

$$\frac{\partial P}{\partial \theta_{\hat{u}}} = \frac{Pr'(e', \hat{u})}{\theta_{\hat{u}}},$$
$$\frac{\partial P}{\partial \theta_{\hat{s}|u}} = \frac{Pr'(e', \hat{s}, u)}{\theta_{\hat{s}|u}} = \frac{Pr'(e', u)}{\theta_{\hat{s}|u}},$$

the last step because $\hat{s}$ already belongs to e'.
G is scale-invariant, so the simplex constraint carries no multiplier. Scaling $\theta_{\hat{u}} \mapsto \alpha\theta_{\hat{u}}$ for all $\hat{u}$ at one deleted edge sends $P \mapsto \alpha P$ (one such factor per term) and $z_{UX} \mapsto \alpha z_{UX}$, hence $z \mapsto \alpha z$, leaving G unchanged; the same holds for the group $\{\theta_{\hat{s}|u}\}_u$. Euler's relation for a degree-0 homogeneous function gives

$$\sum_{\hat{u}} \theta_{\hat{u}} \frac{\partial G}{\partial \theta_{\hat{u}}} = 0, \qquad \sum_{u} \theta_{\hat{s}|u} \frac{\partial G}{\partial \theta_{\hat{s}|u}} = 0.$$

The $\theta_{\hat{s}|u}$ are unconstrained coordinates ($\hat{S}$ being observed, the values $\theta_{\bar{\hat{s}}|u}$ never appear in G), so stationarity there is $\partial G/\partial\theta_{\hat{s}|u} = 0$. The $\theta_{\hat{u}}$ lie on the simplex, so stationarity is $\partial G/\partial\theta_{\hat{u}} = \mu_{UX}$ for a common multiplier; multiplying by $\theta_{\hat{u}}$ and summing, Euler gives $0 = \mu_{UX} \sum_{\hat{u}} \theta_{\hat{u}} = \mu_{UX}$. So stationarity is exactly $\partial G/\partial\theta_{\hat{u}} = \partial G/\partial\theta_{\hat{s}|u} = 0$ at every deleted edge and state. Since $\partial z/\partial\theta_{\hat{s}|u} = (z/z_{UX})\theta_{\hat{u}}$ and $\partial z/\partial\theta_{\hat{u}} = (z/z_{UX})\theta_{\hat{s}|u}$ (states $u = \hat{u}$), the quotient rule gives

$$\frac{\partial G}{\partial \theta_{\hat{s}|u}} = \frac{1}{z^2}\left[\frac{Pr'(e', u)}{\theta_{\hat{s}|u}}\, z \;-\; P\, \frac{z}{z_{UX}}\, \theta_{\hat{u}}\right].$$

Setting this to zero and multiplying through by $z_{UX}\theta_{\hat{s}|u}/z$ gives $Pr'(e', u) z_{UX} = P\theta_{\hat{u}}\theta_{\hat{s}|u}$, i.e. $\theta_{\hat{u}}\theta_{\hat{s}|u} = z_{UX} Pr'(u \mid e')$; in the same way $\partial G/\partial\theta_{\hat{u}} = 0$ gives $\theta_{\hat{u}}\theta_{\hat{s}|u} = z_{UX} Pr'(\hat{u} \mid e')$. So stationarity of $Pr'(e')/z$ is equivalent to

$$\theta_{\hat{u}}\theta_{\hat{s}|u} \;=\; z_{UX}\, Pr'(u \mid e') \;=\; z_{UX}\, Pr'(\hat{u} \mid e') \qquad \text{for every deleted edge and every } u = \hat{u}, \tag{F}$$

which is Equation 14.26 from the proof of Lemma 14.1; it remains to prove (F) $\iff$ (14.8) and (14.9). Two facts recur: $\hat{U}$ is a root, so $Pr'(\hat{u}) = \theta_{\hat{u}}$; and $\hat{S}$ is a leaf whose only parent is U, hence d-separated

from everything else by U, so

$$\begin{aligned} Pr'(\hat{s} \mid u, e' - \hat{S}) \;&=\; Pr'(\hat{s} \mid u) \;=\; \theta_{\hat{s}|u}, \\ \text{hence} \quad Pr'(u, e') \;&=\; Pr'(u, e' - \hat{S})\, \theta_{\hat{s}|u}. \end{aligned} \tag{C}$$

(14.8) and (14.9) imply (F). Fix a deleted edge. Using (14.9) and $Pr'(\hat{u}) = \theta_{\hat{u}}$,

$$\theta_{\hat{u}}\theta_{\hat{s}|u} = \theta_{\hat{u}} \cdot \eta\, Pr'(e' \mid \hat{u}) = \eta\, Pr'(\hat{u}, e') = \eta\, Pr'(e')\, Pr'(\hat{u} \mid e'). \tag{D}$$

Using (14.8) and then (C),

$$\begin{aligned} \theta_{\hat{u}}\theta_{\hat{s}|u} = Pr'(u \mid e' - \hat{S})\, \theta_{\hat{s}|u} &= \frac{Pr'(u, e' - \hat{S})\, \theta_{\hat{s}|u}}{Pr'(e' - \hat{S})} \\ = \frac{Pr'(u, e')}{Pr'(e' - \hat{S})} &= Pr'(u \mid e') \cdot \frac{Pr'(e')}{Pr'(e' - \hat{S})}. \end{aligned} \tag{E}$$

Summing (D) and (E) over the states $u = \hat{u}$, both left sides being z_{UX} by definition and both posteriors summing to 1, gives $z_{UX} = \eta Pr'(e') = Pr'(e')/Pr'(e' - \hat{S})$, so $\eta = [Pr'(e' - \hat{S})]^{-1}$; substituting back into (D) and (E) yields (F).
(F) implies (14.8) and (14.9). The second equality of (F) with $Pr'(\hat{u}) = \theta_{\hat{u}}$, and the first with (C), give

$$\theta_{\hat{u}}\theta_{\hat{s}|u} = z_{UX} \frac{\theta_{\hat{u}}\, Pr'(e' \mid \hat{u})}{Pr'(e')},$$

$$\theta_{\hat{u}}\theta_{\hat{s}|u} = z_{UX} \frac{Pr'(u, e' - \hat{S})\, \theta_{\hat{s}|u}}{Pr'(e')}.$$

Dividing the first by $\theta_{\hat{u}} > 0$ gives $\theta_{\hat{s}|u} = \eta Pr'(e' \mid \hat{u})$ with $\eta = z_{UX}/Pr'(e') > 0$ independent of u, Equation 14.9. Dividing the second by $\theta_{\hat{s}|u} > 0$ gives $\theta_{\hat{u}} = c\, Pr'(u \mid e' - \hat{S})$ with c independent of u; summing over the states and using $\sum_{\hat{u}} \theta_{\hat{u}} = 1$ and $\sum_u Pr'(u \mid e' - \hat{S}) = 1$ gives $c = 1$, Equation 14.8 – at every deleted edge simultaneously, hence for the whole parameterization. □

Problem 14.19. Prove Equation 14.27 on Page 376. That is, let N′ be the network that results from deleting every edge of a Bayesian network N — so that for each deleted edge $U \to X$ a clone $\hat{U}$ (a root, parent of X) and an observed leaf $\hat{S}$ (a child of U) are added, and e' is the original evidence e together with $\hat{s}$ for every added $\hat{S}$. Let

$$\text{ENT} \;=\; -\sum_{\mathbf{x}} Pr'(\mathbf{x} \mid e') \log Pr'(\mathbf{x} \mid e')$$

be the entropy of the distribution $Pr'(\mathbf{X} \mid e')$, where $\mathbf{X}$ is the set of all variables of N′. Show that, because N′ is fully disconnected, this entropy decomposes as

$$\text{ENT} \;=\; -\sum_{X\hat{\mathbf{U}}} \sum_{x\hat{\mathbf{u}}} Pr'(x\hat{\mathbf{u}} \mid e') \log Pr'(x\hat{\mathbf{u}} \mid e'),$$

where X ranges over the variables of the original network N and $\hat{\mathbf{U}}$ are the cloned parents of X in N′.

Solution. Every edge being deleted, N′ splits into one component per variable X of N,

$$C_X \;=\; \{X\} \cup \{\hat{U}_1, \ldots, \hat{U}_k\} \cup \{\hat{S}_1, \ldots, \hat{S}_m\},$$

where $U_1, \ldots, U_k$ are X's parents and $Y_1, \ldots, Y_m$ its children in N: each clone $\hat{U}_i$ is a root whose only child is X, each $\hat{S}_j$ a leaf whose only parent is X (observed at $\hat{s}_j$), so X's neighbours are exactly these and theirs is exactly X. Every variable of N′ lies in exactly one such component – clones and auxiliaries

are created fresh, one per deleted edge, each attached to one original variable – so $\mathbf{X} = \biguplus_X C_X$. By the chain rule Pr' is the product of all CPTs, and each CPT belongs to one component (a family never straddles two, being a node plus its parents), so for a complete instantiation $\mathbf{x} = (c_X)_X$,

$$Pr'(\mathbf{x}) \;=\; \prod_X Pr'(c_X),$$

where $Pr'(c_X)$ is the marginal over C_X – the product of that component's CPTs, itself a normalized distribution. The evidence splits likewise, $e' = \biguplus_X e'_X$, so writing $\lambda_{e'}(\mathbf{x}) = \prod_X \lambda_{e'_X}(c_X)$ and normalizing,

$$\begin{aligned} Pr'(\mathbf{x} \mid e') &= \frac{\prod_X Pr'(c_X)\lambda_{e'_X}(c_X)}{\prod_X Pr'(e'_X)} \\ &= \prod_X Pr'(c_X \mid e'_X) = \prod_X Pr'(c_X \mid e'), \end{aligned}$$

the last equality because C_X is d-separated from the rest by the empty set. Entropy is additive over such blocks: for $P(\mathbf{x}) = \prod_i P_i(w_i)$ over disjoint $W_1, \ldots, W_r$, with $0 \log 0 = 0$,

$$\begin{aligned} -\sum_{\mathbf{x}} P(\mathbf{x}) \log P(\mathbf{x}) &= -\sum_i \sum_{w_i} \Big(\sum_{\mathbf{x} \sim w_i} P(\mathbf{x}) \Big) \log P_i(w_i) \\ &= -\sum_i \sum_{w_i} P_i(w_i) \log P_i(w_i), \end{aligned}$$

the inner sum marginalizing P down to $P_i(w_i)$. With $P = Pr'(\cdot \mid e')$ and blocks C_X, this makes $\mathrm{ENT} = -\sum_X \sum_{c_X} Pr'(c_X \mid e') \log Pr'(c_X \mid e')$. Finally write $c_X = (x, \hat{\mathbf{u}}, \hat{\mathbf{s}})$. Every $\hat{S}_j$ is instantiated by e', so $Pr'(c_X \mid e') = 0$ unless $\hat{\mathbf{s}}$ is the observed instantiation, in which case it equals $Pr'(x\hat{\mathbf{u}} \mid e')$; the vanishing terms contribute nothing under $0 \log 0 = 0$, so the inner sum collapses to one over $x\hat{\mathbf{u}}$. Since $X\hat{\mathbf{U}}$ is exactly the family of X in N′,

$$\mathrm{ENT} \;=\; -\sum_{X\hat{\mathbf{U}}} \sum_{x\hat{\mathbf{u}}} Pr'(x\hat{\mathbf{u}} \mid e') \log Pr'(x\hat{\mathbf{u}} \mid e'),$$

which is Equation 14.27. □

Problem 14.20. Consider a Bayesian network with families $X\mathbf{U}$ and distribution $Pr(\mathbf{X})$, and let $\lambda_e(x)$ denote the evidence indicator of variable X at value x, that is, $\lambda_e(x) = 0$ if evidence e sets X to a value other than x and $\lambda_e(x) = 1$ otherwise. Show that for evidence e with $Pr(e) > 0$, we have $\log Pr(e) = \mathrm{ENT} + \mathrm{AVG}$ where

- $\mathrm{ENT} = -\sum_{\mathbf{x}} Pr(\mathbf{x} \mid e) \log Pr(\mathbf{x} \mid e)$ is the entropy of distribution $Pr(\mathbf{X} \mid e)$.
- $\mathrm{AVG} = \sum_{X\mathbf{U}} \sum_{x\mathbf{u}} Pr(x\mathbf{u} \mid e) \log \lambda_e(x)\theta_{x|\mathbf{u}}$ is a sum of expectations over network parameters.

Solution. Average the pointwise identity $\log Pr(e) = \log Pr(\mathbf{x}, e) - \log Pr(\mathbf{x} \mid e)$ against $Pr(\mathbf{x} \mid e)$. Throughout $p \log q$ is read as 0 whenever $p = 0$, which makes the terms with $\lambda_e(x) = 0$ or $\theta_{x|\mathbf{u}} = 0$ harmless: both force $Pr(x\mathbf{u} \mid e) = 0$. By the chain rule $Pr(\mathbf{x}) = \prod_{X\mathbf{U}} \theta_{x|\mathbf{u}}$ with $x\mathbf{u} \sim \mathbf{x}$, while $\prod_X \lambda_e(x)$ is 1 exactly when $\mathbf{x} \sim e$ and 0 otherwise, and $Pr(\mathbf{x}, e)$ is $Pr(\mathbf{x})$ or 0 accordingly. Hence

$$Pr(\mathbf{x}, e) \;=\; \prod_{X\mathbf{U}} \lambda_e(x)\, \theta_{x|\mathbf{u}}, \qquad x\mathbf{u} \sim \mathbf{x},$$

one factor per variable, so $\log Pr(\mathbf{x}, e) = \sum_{X\mathbf{U}} \log \lambda_e(x)\theta_{x|\mathbf{u}}$ whenever $Pr(\mathbf{x}, e) > 0$. Multiplying the pointwise identity by $Pr(\mathbf{x} \mid e)$ and summing over the $\mathbf{x}$ with $Pr(\mathbf{x} \mid e) > 0$, the left side becomes

$\log Pr(e)$ since $\sum_{\mathbf{x}} Pr(\mathbf{x} \mid e) = 1$, and

$$\begin{aligned}\log Pr(e) \;&=\; \sum_{\mathbf{x}} Pr(\mathbf{x} \mid e) \log Pr(\mathbf{x}, e) \\ &\quad - \sum_{\mathbf{x}} Pr(\mathbf{x} \mid e) \log Pr(\mathbf{x} \mid e).\end{aligned}$$

The second sum is, with its minus sign, the entropy ENT of $Pr(\mathbf{X} \mid e)$. For the first, substitute the parameter product and exchange the two finite sums,

$$\begin{aligned}\sum_{\mathbf{x}} Pr(\mathbf{x} \mid e) \log Pr(\mathbf{x}, e) &= \sum_{\mathbf{x}} Pr(\mathbf{x} \mid e) \sum_{X\mathbf{U}} \log \lambda_e(x)\theta_{x|\mathbf{u}} \\ &= \sum_{X\mathbf{U}} \sum_{\mathbf{x}} Pr(\mathbf{x} \mid e) \log \lambda_e(x)\theta_{x|\mathbf{u}}.\end{aligned}$$

and for a fixed family the summand depends on $\mathbf{x}$ only through $x\mathbf{u}$, so group the complete instantiations accordingly:

$$\begin{aligned}\sum_{\mathbf{x}} Pr(\mathbf{x} \mid e) \log \lambda_e(x)\theta_{x|\mathbf{u}} &= \sum_{x\mathbf{u}} \left(\sum_{\mathbf{x} \sim x\mathbf{u}} Pr(\mathbf{x} \mid e) \right) \log \lambda_e(x)\theta_{x|\mathbf{u}} \\ &= \sum_{x\mathbf{u}} Pr(x\mathbf{u} \mid e) \log \lambda_e(x)\theta_{x|\mathbf{u}},\end{aligned}$$

the inner sum being the family marginal $Pr(x\mathbf{u} \mid e)$. Summing over families, the first sum is AVG, so $\log Pr(e) = \text{ENT} + \text{AVG}$. □

14 Approximate Inference by Stochastic Sampling

Contents

14.1 Exercises 15.1–15.7 246
14.2 Exercises 15.8–15.14 250
14.3 Exercises 15.15–15.21 255
14.4 Exercises 15.22–15.28 259
14.5 Exercises 15.29–15.31 266

14.1 Exercises 15.1–15.7

Problem 15.1. Let N_a be the number of students in a class having age a:

Age a	N_a
18	5
19	15
20	21
21	30
22	20
23	6
24	3

What is the expected age of a student chosen randomly from this class? Formulate this problem by defining a function f where the expected value of the function corresponds to the answer. Compute the variance of the identified function.

Solution. Expected age 20.75 years, variance 1.9075. Take X ranging over the students with $\Pr(X)$ uniform and $f(x)$ the age of student x; the answer is $\mathrm{Ex}(f)$. Grouping the $N = \sum_a N_a = 100$ equiprobable students by age gives $\Pr(f = a) = N_a/N$, so by Equation 15.1

$$\mathrm{Ex}(f) \;=\; \sum_x f(x)\Pr(x) \;=\; \sum_a a\,\frac{N_a}{N} \;=\; \frac{1}{100}\sum_a a\,N_a.$$

Term by term:

a	18	19	20	21	22	23	24	sum
N_a	5	15	21	30	20	6	3	100
aN_a	90	285	420	630	440	138	72	2075

$$\mathrm{Ex}(f) \;=\; \frac{2075}{100} \;=\; 20.75 \text{ years.}$$

By $\mathrm{Va}(f) = \mathrm{Ex}(f^2) - \mathrm{Ex}(f)^2$ (Exercise 15.2):

a	18	19	20	21	22	23	24	sum
a^2N_a	1620	5415	8400	13230	9680	3174	1728	43247

so $\mathrm{Ex}(f^2) = 43247/100 = 432.47$, and

$$\mathrm{Va}(f) \;=\; 432.47 - (20.75)^2 \;=\; 1.9075.$$

Problem 15.2. Show that the variance of function $f(X)$ with respect to distribution $\Pr(X)$ satisfies

$$\mathrm{Va}(f) \;=\; \mathrm{Ex}(f^2) - \mathrm{Ex}(f)^2.$$

Solution. Expand the square in the definition of variance, Equation 15.2, writing $\mu = \mathrm{Ex}(f)$ for the constant $\sum_x f(x)\Pr(x)$ of Equation 15.1:

$$\begin{aligned}\mathrm{Va}(f) &= \sum_x \big(f(x) - \mu\big)^2 \Pr(x)\\ &= \sum_x f(x)^2\Pr(x) - 2\mu\sum_x f(x)\Pr(x) + \mu^2\sum_x \Pr(x)\\ &= \mathrm{Ex}(f^2) - 2\mu^2 + \mu^2 \;=\; \mathrm{Ex}(f^2) - \mathrm{Ex}(f)^2,\end{aligned}$$

since the middle sum is μ by Equation 15.1 and $\sum_x \Pr(x) = 1$.

Problem 15.3. Let μ and σ^2 be the expectation and variance of a function and let v be one of its observed values. Using Chebyshev's inequality (Theorem 15.2), what is the value of ϵ that allows us to state that μ lies in the interval $(v-\epsilon, v+\epsilon)$ with confidence $\geq 95\%$? Compare this with the value

of ϵ based on Theorem 15.3.
Recall the two results. Theorem 15.2 (Chebyshev's inequality): for any $\epsilon > 0$,

$$P(|v - \mu| < \epsilon) \;\geq\; 1 - \frac{\sigma^2}{\epsilon^2}.$$

Theorem 15.3: if the values of the function are normally distributed, then $P(|v - \mu| < \epsilon) = \alpha$ where

$$\epsilon \;=\; \Phi^{-1}\!\left(\frac{1+\alpha}{2}\right)\sigma,$$

with $\Phi(\cdot)$ the CDF of the standard Normal distribution.

Solution. Chebyshev needs $\epsilon = \sqrt{20}\,\sigma \approx 4.472\,\sigma$; Theorem 15.3 needs only $\epsilon = 1.96\,\sigma$. Both theorems bound $P(|v - \mu| < \epsilon)$, and $|v - \mu| < \epsilon$ is exactly the event $\mu \in (v - \epsilon, v + \epsilon)$.
(i) Chebyshev (Theorem 15.2). The confidence is at least $1 - \sigma^2/\epsilon^2$, so

$$\begin{aligned} 1 - \frac{\sigma^2}{\epsilon^2} \;\geq\; 0.95 &\iff \epsilon^2 \;\geq\; 20\,\sigma^2 \\ &\iff \epsilon \;\geq\; 2\sqrt{5}\,\sigma \;\approx\; 4.472\,\sigma. \end{aligned}$$

(ii) Theorem 15.3. With the values normally distributed and $\alpha = 0.95$, $\Phi^{-1}((1+\alpha)/2) = \Phi^{-1}(0.975) = 1.96$, and the confidence is then exactly 0.95 rather than a lower bound:

$$\epsilon \;=\; 1.96\,\sigma.$$

The Normal-based interval is narrower by $\sqrt{20}/1.96 \approx 2.28$.

Problem 15.4. Let f be a function with expectation μ and variance $\sigma^2 = 4$. Suppose we wish to estimate the expectation μ using Monte Carlo simulation (Equation 15.5) and a sample of size n. How large should n be if we want to guarantee that expectation μ falls in the interval $(\mathrm{Av}_n(f) - 1,\ \mathrm{Av}_n(f) + 1)$ with confidence $\geq 99\%$? Use Chebyshev's inequality (Theorem 15.2).
Here $\mathrm{Av}_n(f) = \frac{1}{n}\sum_{i=1}^{n} f(x_i)$ is the sample mean of Equation 15.5.

Solution. $n = 400$. Chebyshev is applied not to f but to the sample mean, which is itself a function of the sample space and by Theorem 15.5 has expectation μ and variance $\sigma^2/n = 4/n$. Theorem 15.2 with these two numbers gives, for every $\epsilon > 0$,

$$P\big(|\mathrm{Av}_n(f) - \mu| < \epsilon\big) \;\geq\; 1 - \frac{4}{n\epsilon^2},$$

and as in Exercise 15.3 the event on the left is $\mu \in (\mathrm{Av}_n(f) - \epsilon,\ \mathrm{Av}_n(f) + \epsilon)$. Setting $\epsilon = 1$ and demanding confidence 0.99,

$$1 - \frac{4}{n} \;\geq\; 0.99 \quad\Longleftrightarrow\quad \frac{4}{n} \;\leq\; 0.01 \quad\Longleftrightarrow\quad n \;\geq\; 400.$$

Problem 15.5. Prove the law of large numbers (Theorem 15.6): let $\mathrm{Av}_n(f)$ be a sample mean where the function f has expectation μ. For every $\epsilon > 0$,

$$\lim_{n\to\infty} P\big(|\mathrm{Av}_n(f) - \mu| \leq \epsilon\big) \;=\; 1.$$

Here $\mathrm{Av}_n(f) = \frac{1}{n}\sum_{i=1}^{n} f(x_i)$ (Equation 15.5), where $x_1, \ldots, x_n$ is a random sample drawn independently from the distribution $\Pr(X)$.

Solution. Apply Chebyshev to the sample mean and squeeze. Since X is a finite set of discrete variables, $\sigma^2 = \mathrm{Va}(f)$ is finite; the sample mean is itself a function of the sample space $(x_1, \ldots, x_n)$ and by Theorem 15.5 (the x_i drawn independently) has expectation μ and variance σ^2/n. Hence Theorem 15.2 applied to it, with monotonicity of probability, gives for every $\epsilon > 0$ and every n

$$1 - \frac{\sigma^2}{n\epsilon^2} \;\le\; P\big(|\mathrm{Av}_n(f) - \mu| < \epsilon\big) \;\le\; P\big(|\mathrm{Av}_n(f) - \mu| \le \epsilon\big) \;\le\; 1.$$

With ϵ fixed the left end tends to 1, so the squeeze gives $\lim_{n\to\infty} P(|\mathrm{Av}_n(f) - \mu| \le \epsilon) = 1$. ■

Problem 15.6. Consider the Bayesian network in Figure 15.1 and the parameter $\theta_{\bar{a}}$ representing the probability of $A =$ false (i.e., it is not winter). For each of the following values of this parameter, .01, .4, and .99, do the following:

- (a) Compute the probability $\Pr(d, e)$: wet grass and slippery road.
- (b) Estimate $\Pr(d, e)$ using direct sampling with sample sizes ranging from $n = 100$ to $n = 15{,}000$.
- (c) Generate a plot with n on the x-axis and the exact value of $\Pr(d, e)$ and the estimate for $\Pr(d, e)$ on the y-axis.
- (d) Generate a plot with n on the x-axis and the exact variance of the estimate for $\Pr(d, e)$ and the sample variance on the y-axis.

The network of Figure 15.1 has five binary variables: A (Winter?), B (Sprinkler?), C (Rain?), D (Wet Grass?), E (Slippery Road?), with edges

$$A \to B, \quad A \to C, \quad B \to D, \quad C \to D, \quad C \to E.$$

Its CPTs are:

A	Θ_A
true	.6
false	.4

A	B	$\Theta_{B\|A}$
true	true	.2
true	false	.8
false	true	.75
false	false	.25

A	C	$\Theta_{C\|A}$
true	true	.8
true	false	.2
false	true	.1
false	false	.9

B	C	D	$\Theta_{D\|BC}$
true	true	true	.95
true	true	false	.05
true	false	true	.9
true	false	false	.1
false	true	true	.8
false	true	false	.2
false	false	true	0
false	false	false	1

C	E	$\Theta_{E\|C}$
true	true	.7
true	false	.3
false	true	0
false	false	1

In this exercise Θ_A is varied: $\theta_{\bar{a}} \in \{.01, .4, .99\}$ and $\theta_a = 1 - \theta_{\bar{a}}$. Lowercase a, b, c, d, e denote the value true and $\bar{a}, \bar{b}, \bar{c}, \bar{d}, \bar{e}$ the value false.

Solution. (a) $\Pr(d,e) = .4648 - .400925\,\theta_{\bar a}$, a decreasing linear function of $\theta_{\bar a}$. Eliminate E and then B from the network factorization $\Pr(A)\Pr(B|A)\Pr(C|A)\Pr(D|B,C)\Pr(E|C)$: every $C=\bar c$ term vanishes since $\Pr(e|\bar c)=0$, leaving the constant factor $\Pr(e|c)=.7$, so with $g(A)=\sum_B \Pr(B|A)\Pr(d|B,c)$,

$$\begin{aligned} g(a) &= (.2)(.95)+(.8)(.8) = .83,\\ g(\bar a) &= (.75)(.95)+(.25)(.8) = .9125,\\ \Pr(d,e) &= .7\big[\theta_a(.8)(.83)+\theta_{\bar a}(.1)(.9125)\big]\\ &= .7\big[.664 - .57275\,\theta_{\bar a}\big] \;=\; .4648 - .400925\,\theta_{\bar a}, \end{aligned}$$

using $\theta_a = 1-\theta_{\bar a}$. At the three requested values:

$\theta_{\bar a}$	$\Pr(d,e)$ exact
.01	.46079075
.4	.30443
.99	.06788425

(b) Algorithm 45 with $\alpha : d \wedge e$, each trial simulated in the order A,B,C,D,E (Algorithm 44) and scored by $\breve\alpha$ of Definition 15.1, returns $\mathrm{Av}_n(\breve\alpha)$ of Equation 15.7. One run at selected sample sizes:

n	$\theta_{\bar a}=.01$	$\theta_{\bar a}=.4$	$\theta_{\bar a}=.99$
100	.3700	.2700	.0800
500	.4260	.2960	.0640
1000	.4590	.3010	.0550
2500	.4648	.3148	.0668
5000	.4624	.3058	.0686
10000	.4631	.3052	.0693
15000	.4619	.3080	.0691
exact	.4607907	.3044300	.0678842

(c) Each panel plots the horizontal line $\Pr(d,e)$ against $n \mapsto \mathrm{Av}_n(\breve\alpha)$, which fluctuates for small n (absolute errors .091, .034, .012 at $n=100$) and settles onto the line by Theorem 15.6. Superimpose the funnel $\Pr(d,e) \pm 1.96\sqrt{\Pr(\alpha)\Pr(\neg\alpha)/n}$, since by Corollary 4 the estimate has standard deviation $\sqrt{\Pr(\alpha)\Pr(\neg\alpha)/n}$ and Theorem 15.7 puts roughly 95% of the trajectory inside; at $n=15{,}000$ its half-widths are .0080, .0074, .0040, and all three estimates above fall inside.

(d) By Corollary 4 the exact variance of the estimate is

$$\mathrm{Va}\big(\mathrm{Av}_n(\breve\alpha)\big) = \frac{\Pr(\alpha)\Pr(\neg\alpha)}{n},$$

and its empirical counterpart is the sample variance of Equation 15.6 divided by n. Since $\breve\alpha(x_i)\in\{0,1\}$, writing $\hat p = \mathrm{Av}_n(\breve\alpha)$ gives $\sum_i \breve\alpha(x_i)^2 = n\hat p$, so by the one-pass identity of Exercise 15.7,

$$S_n^2(\breve\alpha) = \frac{n\hat p - n\hat p^2}{n-1} = \frac{n}{n-1}\,\hat p(1-\hat p), \qquad \frac{S_n^2(\breve\alpha)}{n} = \frac{\hat p(1-\hat p)}{n-1}.$$

Plotting both against n for the same run:

n	exact .01	sample .01	exact .4	sample .4	exact .99	sample .99
100	2.485e-3	2.354e-3	2.118e-3	1.991e-3	6.328e-4	7.434e-4
500	4.969e-4	4.900e-4	4.235e-4	4.176e-4	1.266e-4	1.201e-4
1000	2.485e-4	2.486e-4	2.118e-4	2.106e-4	6.328e-5	5.203e-5
2500	9.939e-5	9.954e-5	8.470e-5	8.632e-5	2.531e-5	2.494e-5
5000	4.969e-5	4.973e-5	4.235e-5	4.247e-5	1.266e-5	1.278e-5
10000	2.485e-5	2.487e-5	2.118e-5	2.121e-5	6.328e-6	6.450e-6
15000	1.656e-5	1.657e-5	1.412e-5	1.421e-5	4.218e-6	4.287e-6

Column "exact t" is $\Pr(\alpha)\Pr(\neg\alpha)/n$ at $\theta_{\bar a}=t$; column "sample t" is $S_n^2(\breve\alpha)/n$ from the run of part (b). Both decay like $1/n$ and are indistinguishable beyond $n \approx 2000$.

Problem 15.7. Show that the sample variance of function $f(X)$ and sample $x_1, \ldots, x_n$ can be computed as follows:

$$S_n^2(f) \;=\; \frac{1}{n-1}\,T,$$

where

$$T \;=\; \sum_{i=1}^{n}\big(f(x_i) - \mathrm{Av}_n(f)\big)^2 \;=\; \sum_{i=1}^{n} f(x_i)^2 \;-\; \frac{1}{n}\left(\sum_{i=1}^{n} f(x_i)\right)^2.$$

Note that the first form of T given in this chapter suggests computation by a two-pass algorithm: compute the mean in one pass and compute the differences squared in another pass. The second form of T suggests computation by a one-pass algorithm that simply accumulates the sums of $f(x_i)^2$ and $f(x_i)$.

Solution. Expand the square. Equation 15.6 already defines $S_n^2(f)$ as $1/(n-1)$ times the first expression for T, so only the identity between the two forms of T is at issue. Put $f_i = f(x_i)$, $S = \sum_i f_i$, $Q = \sum_i f_i^2$, and $m = \mathrm{Av}_n(f) = S/n$ (Equation 15.5), a constant in i. Then

$$\begin{aligned} T &= \sum_{i=1}^{n}(f_i - m)^2 = \sum_{i=1}^{n} f_i^2 - 2m\sum_{i=1}^{n} f_i + \sum_{i=1}^{n} m^2 \\ &= Q - 2mS + nm^2 = Q - \frac{2S^2}{n} + \frac{S^2}{n} = Q - \frac{S^2}{n}, \end{aligned}$$

the third sum having n identical terms m^2; and $Q - S^2/n$ is the claimed second form. It needs only the running totals S and Q, so one pass with $O(1)$ memory reports $\mathrm{Av}_n(f) = S/n$ and $S_n^2(f) = (Q - S^2/n)/(n-1)$.

14.2 Exercises 15.8–15.14

Problem 15.8. Suppose we want to simulate a Bayesian network N that has been conditioned on evidence e. Suppose further that we are given cluster marginals $Pr(C_i|e)$ and separator marginals $Pr(S_{ij}|e)$ computed by running the jointree algorithm on network N and evidence e (here $(T, \mathbf{C})$ is a jointree for N with clusters C_i, and $S_{ij} = C_i \cap C_j$ is the separator of edge i–j of T). Give an efficient algorithm for simulating the network N conditioned on e given these cluster and separator marginals. That is, show how we can efficiently generate a sequence of independent network instantiations $\mathbf{x}^1, \ldots, \mathbf{x}^n$ where the probability of generating instantiation $\mathbf{x}^i$ is $Pr(\mathbf{x}^i|e)$.

Solution. Walk the jointree outward from a root cluster, sampling one whole cluster at a time from a conditional built out of the given marginals — the jointree analogue of Algorithm 44, SIMULATE BN. Root T at C_r and list the clusters in a prefix order $C_{i_1} = C_r, C_{i_2}, \ldots, C_{i_m}$ (any BFS or DFS from C_r), writing $S_k = C_{i_k} \cap C_{\pi(k)}$ for the separator to the tree-parent and $V_k = C_{i_1} \cup \cdots \cup C_{i_k}$, so $V_m = \mathbf{X}$ as the clusters cover all network variables. Once, store $Pr(C_r|e)$ and for $k = 2, \ldots, m$ the entrywise quotient

$$Q_k(C_{i_k} \setminus S_k \mid S_k) \;=\; \frac{Pr(c_{i_k}|e)}{Pr(s_k|e)},$$

the division the jointree algorithm already performs. Then each sample is:

1. Sample c_{i_1} from $Pr(C_r|e)$.
2. For $k = 2, \ldots, m$: read off the value s_k the already-sampled variables give S_k — they all lie in $C_{\pi(k)}$, instantiated earlier — and sample t_k of $C_{i_k} \setminus S_k$ from $Q_k(\,\cdot \mid s_k)$.
3. Return the instantiation $\mathbf{x}$ assembled from $c_{i_1}, t_2, \ldots, t_m$.

Correctness: deleting the tree edge above C_{i_k} splits the clusters into a root side $\mathbf{A} \supseteq V_{k-1}$ and a far side $\mathbf{B} \supseteq C_{i_k}$ with $\mathbf{A} \cap \mathbf{B} = S_k$ (running intersection), and $Pr(\mathbf{x}|e) = \prod_i \psi_i(c_i)$ factorizes over the clusters (chain rule (4.2) times evidence indicators over $Pr(e)$, every family and evidence variable sitting in

some cluster), so grouping by side gives $Pr(\mathbf{x}|e) = f(\mathbf{a})g(\mathbf{b})$ and hence $V_{k-1} \setminus S_k \perp C_{i_k} \setminus S_k \mid S_k$ in $Pr(\cdot|e)$; also $C_{i_k} \cap V_{k-1} = S_k$, so no variable is re-sampled. Inducting on k, with s_k part of v_{k-1},

$$Pr(v_{k-1}|e) \cdot \frac{Pr(s_k t_k|e)}{Pr(s_k|e)} = Pr(v_{k-1}|e)\, Pr(t_k \mid v_{k-1}, e) = Pr(v_k|e),$$

no division by zero occurring since $Pr(s_k|e) \geq Pr(v_{k-1}|e) > 0$ for any instantiation actually produced; at $k = m$ this is $Pr(\mathbf{x}|e)$, fresh random numbers make the $\mathbf{x}^i$ independent, and every sample is consistent with e since $Pr(c_i|e) = 0$ at cluster instantiations contradicting e.
Preprocessing costs $O(\sum_i \exp(|C_i|))$, the order of the jointree itself; with each conditional turned into cumulative sums or an alias table the m draws are $O(1)$ apiece, so n samples cost $O(\sum_i \exp(|C_i|) + nm)$ with nothing discarded.

Problem 15.9. Recall Algorithm 44, SIMULATE BN: given a Bayesian network N over variables $\mathbf{X}$ inducing distribution $Pr(\mathbf{X})$, it fixes a total order $\pi = X_1, \ldots, X_n$ of the network variables in which parents appear before their children, starts with the trivial instantiation, and then for $i = 1, \ldots, n$ lets $\mathbf{u}_i$ be the value of X_i's parents in the instantiation built so far, samples a value x_i for X_i from $Pr(X_i|\mathbf{u}_i)$, and appends it. It returns the complete instantiation $\mathbf{x} = x_1, \ldots, x_n$.
Show the following for an instantiation $\mathbf{x}$ returned by Algorithm 44, SIMULATE BN:

- (a) The partial instantiation $x_1, \ldots, x_k$ of $\mathbf{x}$ is generated with probability $Pr(x_1, \ldots, x_k)$, where X_i is the variable at position i in the used order π.
- (b) The value x assigned to variable X by instantiation $\mathbf{x}$ is generated with probability $Pr(x)$.
- (c) The instantiation $\mathbf{c}$ assigned to variables $\mathbf{C} \subseteq \mathbf{X}$ by $\mathbf{x}$ is generated with probability $Pr(\mathbf{c})$.

Hint: You can use (a) to prove (b), and use (b) to prove (c).

Solution. (a) Both sides equal $\prod_{i=1}^{k} \theta_{x_i|\mathbf{u}_i}$, where $\mathbf{u}_i$ is the instantiation that $x_1, \ldots, x_{i-1}$ gives the parents $\mathbf{U}_i$ of X_i. Since π places parents before children, at iteration i the partial instantiation already fixes $\mathbf{U}_i$ and step 7 draws x_i with probability $\theta_{x_i|\mathbf{u}_i}$; successive draws being independent given the history,

$$P_{\text{alg}}(x_1, \ldots, x_k) \;=\; \prod_{i=1}^{k} \theta_{x_i|\mathbf{u}_i}.$$

For the distribution, $k = n$ is the chain rule (4.2); smaller k follows by summing out trailing variables, X_n having no children (a child would follow it in π) and so occurring in no $\mathbf{u}_i$ with $i < n$:

$$\begin{aligned} Pr(x_1, \ldots, x_{n-1}) &= \sum_{x_n} \prod_{i=1}^{n} \theta_{x_i|\mathbf{u}_i} \\ &= \left(\prod_{i=1}^{n-1} \theta_{x_i|\mathbf{u}_i} \right) \sum_{x_n} \theta_{x_n|\mathbf{u}_n} \;=\; \prod_{i=1}^{n-1} \theta_{x_i|\mathbf{u}_i}, \end{aligned}$$

every CPT column summing to 1; repeat down to position $k + 1$.
(b) Let $X = X_k$. The returned instantiation assigns x to X_k exactly when the first k iterations produce one of the sequences $x_1, \ldots, x_{k-1}, x$; these are mutually exclusive and no later iteration changes the value of X_k. So by (a) and marginalization over $X_1, \ldots, X_{k-1}$,

$$P_{\text{alg}}(X_k = x) = \sum_{x_1, \ldots, x_{k-1}} Pr(x_1, \ldots, x_{k-1}, x) \;=\; Pr(x).$$

(c) The same argument with k the largest position in π occupied by a variable of $\mathbf{C}$, so that $\mathbf{C} \subseteq \{X_1, \ldots, X_k\}$. Writing $x_1, \ldots, x_k \sim \mathbf{c}$ for compatibility, those events are mutually exclusive, undisturbed by later iterations, and their disjunction is $\mathbf{c}$, so by (a)

$$P_{\text{alg}}(\mathbf{c}) = \sum_{x_1, \ldots, x_k \sim \mathbf{c}} Pr(x_1, \ldots, x_k) \;=\; Pr(\mathbf{c}).$$

Problem 15.10. Prove Theorem 15.9.
Recall the setting. We have a distribution $Pr(\mathbf{X}, \mathbf{Y})$ over disjoint sets of variables $\mathbf{X}$ and $\mathbf{Y}$ and an event α. By Definition 15.2, the Rao-Blackwell function for event α and distribution $Pr(\mathbf{X}, \mathbf{Y})$, denoted $\ddot{\alpha}(\mathbf{Y})$, maps each instantiation $\mathbf{y}$ into $[0, 1]$ as follows:

$$\ddot{\alpha}(\mathbf{y}) \stackrel{\text{def}}{=} Pr(\alpha|\mathbf{y}).$$

Theorem 15.9 states that the expectation and variance of the Rao-Blackwell function $\ddot{\alpha}(\mathbf{Y})$ with respect to the distribution $Pr(\mathbf{Y})$ are

$$Ex(\ddot{\alpha}) = Pr(\alpha),$$

$$Va(\ddot{\alpha}) = \sum_{\mathbf{y}} Pr(\alpha|\mathbf{y})^2 \, Pr(\mathbf{y}) - Pr(\alpha)^2.$$

Solution. Both halves are the definitions plus one case analysis (setting $\ddot{\alpha}(\mathbf{y}) = 0$ where $Pr(\mathbf{y}) = 0$, such terms carrying weight 0 in every sum). For the expectation, Equation 15.1 and then the definition of conditional probability give

$$\begin{aligned} Ex(\ddot{\alpha}) &= \sum_{\mathbf{y}} \ddot{\alpha}(\mathbf{y}) \, Pr(\mathbf{y}) = \sum_{\mathbf{y}} Pr(\alpha|\mathbf{y}) \, Pr(\mathbf{y}) \\ &= \sum_{\mathbf{y}} Pr(\alpha \wedge \mathbf{y}) = Pr(\alpha), \end{aligned}$$

the last step being case analysis on $\mathbf{Y}$, whose events $\alpha \wedge \mathbf{y}$ are mutually exclusive with disjunction α. Then $Va(f) = Ex(f^2) - Ex(f)^2$ (Exercise 15.2) with $f = \ddot{\alpha}$ gives

$$\begin{aligned} Va(\ddot{\alpha}) &= Ex(\ddot{\alpha}^2) - Ex(\ddot{\alpha})^2 \\ &= \sum_{\mathbf{y}} Pr(\alpha|\mathbf{y})^2 \, Pr(\mathbf{y}) - Pr(\alpha)^2. \end{aligned}$$

Problem 15.11. Given a Bayesian network over variables $\mathbf{X}$ and $\mathbf{Y}$ where $\mathbf{X} \cap \mathbf{Y} = \emptyset$ and $\mathbf{Y}$ is a loop-cutset, show how $Pr(\mathbf{z}|\mathbf{y})$ can be computed in time polynomial in the network size where $\mathbf{Z} \subseteq \mathbf{X}$.
Here $\mathbf{y}$ is an instantiation of all the cutset variables $\mathbf{Y}$, $\mathbf{z}$ is an instantiation of $\mathbf{Z}$, and, by Definition 8.1, a set of nodes $\mathbf{C}$ is a loop-cutset for a Bayesian network N if removing the edges outgoing from the nodes $\mathbf{C}$ renders the network a polytree.

Solution. Hard-code $\mathbf{y}$ into the CPTs to get a polytree $N_{\mathbf{y}}$, then answer two probability-of-evidence queries on it and divide:

$$Pr(\mathbf{z}|\mathbf{y}) = \frac{Pr(\mathbf{z}, \mathbf{y})}{Pr(\mathbf{y})} = \frac{Pr_{\mathbf{y}}(\mathbf{z}, \mathbf{y})}{Pr_{\mathbf{y}}(\mathbf{y})} \qquad (Pr(\mathbf{y}) > 0).$$

Delete from G every edge outgoing from a node of $\mathbf{Y}$; the result $G_{\mathbf{y}}$ is a polytree by Definition 8.1, and the nodes of $\mathbf{Y}$, keeping their incoming edges, become leaves. Splitting the parents $\mathbf{U}$ of a node V as $\mathbf{U}_X \cup \mathbf{U}_Y$ into those outside and inside $\mathbf{Y}$, the parents of V in $G_{\mathbf{y}}$ are $\mathbf{U}_X$, so give V the CPT

$$\theta^{\mathbf{y}}_{v|\mathbf{u}_X} \stackrel{\text{def}}{=} \theta_{v|\mathbf{u}_X, \mathbf{u}_Y},$$

with $\mathbf{u}_Y$ the value $\mathbf{y}$ assigns $\mathbf{U}_Y$ — legitimate, since for fixed $\mathbf{u}_X$ these numbers sum to 1 over v — at one scan per CPT, so building $N_{\mathbf{y}}$ is linear in the size of N. For every instantiation $\mathbf{x}$ of $\mathbf{X}$, the chain rule (4.2) in $N_{\mathbf{y}}$ and then in N gives

$$Pr_{\mathbf{y}}(\mathbf{x}, \mathbf{y}) = \prod_V \theta^{\mathbf{y}}_{v|\mathbf{u}_X} = \prod_V \theta_{v|\mathbf{u}} = Pr(\mathbf{x}, \mathbf{y}),$$

the middle step legitimate precisely because the instantiation assigns $\mathbf{Y}$ the values $\mathbf{y}$ hard-coded into the CPTs. Summing over the $\mathbf{x}$ compatible with $\mathbf{z}$, and then over all $\mathbf{x}$, yields the two numerators above as evidence queries with evidence $\mathbf{z}, \mathbf{y}$ and $\mathbf{y}$.
The polytree algorithm of Section 7.5.4 — Algorithm 12, FE, with the elimination tree mirroring the polytree — returns the marginals $Pr(C_i, e)$ in $O(n \exp(k))$ time and space, k the maximum number of parents, and summing any one of them over its variables gives $Pr(e)$. A node with k parents already occupies $\Theta(\exp(k))$ entries of the specification, so $O(n \exp(k))$ is linear in the network size, and two runs plus a division leave it there.

Problem 15.12. Given a Bayesian network over variables $\mathbf{X}$ and containing evidence e on variables $\mathbf{E}$, show how we can construct a Gibbs sampler for the distribution $Pr(\mathbf{Y}|e)$, $\mathbf{Y} \subseteq \mathbf{X} \setminus \mathbf{E}$, assuming that we can compute $Pr(\mathbf{y}, e)$ for each instantiation $\mathbf{y}$ of variables $\mathbf{Y}$. That is, we want a Gibbs sampler that allows us to compute expectations with respect to the distribution $Pr(\mathbf{Y}|e)$.
For reference, Definition 15.6 says that, given a distribution $Pr(\mathbf{V})$ over m variables, the corresponding Gibbs transition matrix is

$$P(\mathbf{V}_i = \mathbf{v}' \mid \mathbf{V}_{i-1} = \mathbf{v}) = \begin{cases} 0, & \text{(i)} \\ \dfrac{1}{m} Pr(s'|\mathbf{v} - S), & \text{(ii)} \\ \dfrac{1}{m} \displaystyle\sum_{S \in \mathbf{V}} Pr(s_{\mathbf{v}}|\mathbf{v} - S), & \text{(iii)} \end{cases}$$

where case (i) applies when $\mathbf{v}$ and $\mathbf{v}'$ disagree on more than one variable; case (ii) when they disagree on a single variable S, which has value s' in $\mathbf{v}'$; and case (iii) when $\mathbf{v} = \mathbf{v}'$, with $s_{\mathbf{v}}$ the value of S in $\mathbf{v}$.

Solution. Feeding $Pr(\mathbf{Y}|e)$ — a distribution over the $m = |\mathbf{Y}|$ variables $\mathbf{Y}$ — to Definition 15.6, the only quantities the transition matrix needs are the full conditionals $Pr(S \mid \mathbf{y} - S, e)$ for $S \in \mathbf{Y}$, and each is one normalization away from the oracle: $\mathbf{y} - S, s$ is a complete instantiation of $\mathbf{Y}$ for every state s of S, so $Pr(\mathbf{y} - S, s, e)$ is available and case analysis on S gives

$$Pr(s \mid \mathbf{y} - S, e) = \frac{Pr(\mathbf{y} - S,\, s,\, e)}{\sum_{s'} Pr(\mathbf{y} - S,\, s',\, e)} \;=\; \eta\, Pr(\mathbf{y} - S,\, s,\, e).$$

One full conditional thus costs $|S|$ oracle calls and no other inference; both $Pr(e)$ and $Pr(\mathbf{y} - S, e)$ cancel, so unnormalized values suffice. The sampler is then Algorithm 48, GIBBS NEXT STATE, with the chain-rule computation of Theorem 15.17 replaced by oracle calls. Given a current state $\mathbf{y}$:

1. $S \leftarrow$ a variable chosen uniformly at random from $\mathbf{Y}$.
2. For each state s of S: $P(s) \leftarrow Pr(\mathbf{y} - S,\, s,\, e)$.
3. $\eta \leftarrow \sum_s P(s)$; set $P(s) \leftarrow P(s)/\eta$ for each s.
4. $s \leftarrow$ a value of S sampled from the distribution $P(S)$.
5. Return the instantiation $\mathbf{y} - S,\, s$.

Start from any $\mathbf{y}^1$ with $Pr(\mathbf{y}^1, e) > 0$ and iterate. Step 1 chooses S with probability $1/m$ and step 4 draws s with probability $Pr(s|\mathbf{y} - S, e)$, so the next state $\mathbf{y}'$ arises with probability

- (i) 0 when $\mathbf{y}'$ and $\mathbf{y}$ differ on more than one variable, only the chosen S being allowed to change;
- (ii) $\frac{1}{m} Pr(s'|\mathbf{y} - S, e)$ when they differ only on S, that being the only choice at step 1 that produces the transition;
- (iii) $\frac{1}{m} \sum_{S \in \mathbf{Y}} Pr(s_{\mathbf{y}}|\mathbf{y} - S, e)$ when $\mathbf{y}' = \mathbf{y}$, since any S can leave the state unchanged.

This is the Gibbs transition matrix of Definition 15.6 for $Pr(\mathbf{Y}|e)$. By Theorem 15.16 the chain satisfies detailed balance with respect to $Pr(\mathbf{Y}|e)$, which is therefore stationary — and, if $Pr(\mathbf{y}|e) > 0$ for every $\mathbf{y}$ so that the chain is irreducible, uniquely so. Under that condition Theorem 15.15 gives, for any $f(\mathbf{Y})$,

$$\lim_{n \to \infty} Av_n(f) = \sum_{\mathbf{y}} f(\mathbf{y})\, Pr(\mathbf{y}|e),$$

the expectation with respect to $Pr(\mathbf{Y}|e)$ as required.

Problem 15.13. Show that the variance of Theorem 15.9 ranges between zero and $Pr(\alpha)Pr(\neg\alpha)$. State a condition under which the variance reduces to zero and a condition under which it reduces to $Pr(\alpha)Pr(\neg\alpha)$.
Recall that the variance in question is that of the Rao-Blackwell function $\ddot{\alpha}(\mathbf{Y})$ of Definition 15.2, namely $\ddot{\alpha}(\mathbf{y}) = Pr(\alpha|\mathbf{y})$, with respect to the distribution $Pr(\mathbf{Y})$; by Theorem 15.9 it equals

$$Va(\ddot{\alpha}) \;=\; \sum_{\mathbf{y}} Pr(\alpha|\mathbf{y})^2\, Pr(\mathbf{y}) \;-\; Pr(\alpha)^2.$$

Solution. The variance is 0 exactly when α is independent of $\mathbf{Y}$ and $Pr(\alpha)Pr(\neg\alpha)$ exactly when $\mathbf{Y}$ determines the truth of α. Write $f(\mathbf{y}) = \ddot{\alpha}(\mathbf{y}) = Pr(\alpha|\mathbf{y})$, so that $Ex(f) = Pr(\alpha)$ by Theorem 15.9 and $0 \le f \le 1$, f being a probability; "every $\mathbf{y}$" below means every $\mathbf{y}$ with $Pr(\mathbf{y}) > 0$.
(i) Lower end. $Va(f) = \sum_{\mathbf{y}}(f(\mathbf{y}) - Ex(f))^2 Pr(\mathbf{y}) \ge 0$, a weighted sum of squares, with equality if and only if every summand vanishes:

$$Pr(\alpha|\mathbf{y}) \;=\; Ex(f) \;=\; Pr(\alpha) \quad \text{for every } \mathbf{y},$$

that is, the event α is independent of the sampled variables $\mathbf{Y}$.
(ii) Upper end. From $0 \le f \le 1$ we get $f(\mathbf{y})^2 \le f(\mathbf{y})$, hence $Ex(f^2) \le Ex(f) = Pr(\alpha)$, and $Va(f) = Ex(f^2) - Ex(f)^2$ (Exercise 15.2) gives

$$Va(\ddot{\alpha}) \;\le\; Pr(\alpha) - Pr(\alpha)^2 \;=\; Pr(\alpha)Pr(\neg\alpha),$$

the direct sampling variance of Theorem 15.4. Equality requires $Ex(f^2) = Ex(f)$, i.e. $\sum_{\mathbf{y}} f(\mathbf{y})(1 - f(\mathbf{y}))Pr(\mathbf{y}) = 0$; each summand being nonnegative, this holds if and only if

$$Pr(\alpha|\mathbf{y}) \in \{0, 1\} \quad \text{for every } \mathbf{y},$$

that is, $\mathbf{y}$ already settles whether α holds — as it does whenever α mentions only variables in $\mathbf{Y}$, $\ddot{\alpha}$ then coinciding with the direct sampling function $\breve{\alpha}$.

Problem 15.14. Let $Pr(\mathbf{X})$ be a distribution that is hard to sample from. Let $Pr'(\mathbf{X})$ be another distribution that is easy to sample from, where $Pr(\mathbf{x}) \le c \cdot Pr'(\mathbf{x})$ for all instantiations $\mathbf{x}$ and some constant $c > 0$. The method of rejection sampling performs the following steps to generate a sample of size n from distribution $Pr(\mathbf{X})$:
Repeat n times:

1. Sample an instantiation $\mathbf{x}$ from distribution $Pr'(\mathbf{X})$.
2. Accept $\mathbf{x}$ with probability $Pr(\mathbf{x}) \,/\, \big(c \cdot Pr'(\mathbf{x})\big)$.
3. If $\mathbf{x}$ is not accepted (i.e., rejected), go to Step 1.

We can show that the accepted instantiations represent a random sample from distribution $Pr(\mathbf{X})$.
Consider now the algorithm given in Section 15.5 for computing a conditional probability $Pr(\alpha|\beta)$: generate a sample $\mathbf{x}^1, \ldots, \mathbf{x}^n$ of size n from the network distribution, let c_1 be the number of instantiations in the sample at which $\gamma = \alpha \wedge \beta$ is true and c_2 the number at which β is true, and return $Av_n(\breve{\gamma})/Av_n(\breve{\beta}) = c_1/c_2$. Show how this algorithm can be formulated as an instance of rejection sampling by specifying the corresponding distribution Pr' and constant c.

Solution. Take $Pr'(\mathbf{X}) = Pr(\mathbf{X})$, the network's own distribution, which Algorithm 44 samples in time linear in the network size (Theorem 15.1), and $c = 1/Pr(\beta)$. The schema's hard target, written $\widehat{Pr}$ to keep the names apart from the book's Pr, is $\widehat{Pr}(\mathbf{X}) = Pr(\mathbf{X}|\beta)$.
A complete instantiation $\mathbf{x}$ entails β or entails $\neg\beta$, so with $\breve{\beta}(\mathbf{x}) \in \{0, 1\}$ the direct sampling function

of Definition 15.1,

$$\widehat{Pr}(\mathbf{x}) = \frac{Pr(\mathbf{x} \wedge \beta)}{Pr(\beta)} = \frac{\breve{\beta}(\mathbf{x})\, Pr(\mathbf{x})}{Pr(\beta)} \;\leq\; \frac{Pr(\mathbf{x})}{Pr(\beta)} \;=\; c \cdot Pr'(\mathbf{x}),$$

which is the domination the schema requires. Substituting into step 2, for $Pr'(\mathbf{x}) > 0$,

$$\frac{\widehat{Pr}(\mathbf{x})}{c \cdot Pr'(\mathbf{x})} = \frac{\breve{\beta}(\mathbf{x})\, Pr(\mathbf{x}) \,/\, Pr(\beta)}{\big(1/Pr(\beta)\big)\, Pr(\mathbf{x})} = \breve{\beta}(\mathbf{x}),$$

so both unknowns — $Pr(\beta)$ and the normalization of $\widehat{Pr}$ — cancel and the test is deterministic: accept $\mathbf{x}$ iff β is true at $\mathbf{x}$. The accepted instantiations are thus an independent sample from $Pr(\mathbf{X}|\beta)$, and $Pr(\alpha|\beta) = Ex(\breve{\alpha})$ with respect to $Pr(\cdot|\beta)$ (Theorem 15.4 applied to that distribution), so it is estimated by the sample mean of $\breve{\alpha}$ over them; an accepted instantiation satisfies β, so those also satisfying α are exactly those satisfying $\gamma = \alpha \wedge \beta$:

$$\frac{\#\{\text{accepted } \mathbf{x}^i \text{ at which } \alpha \text{ holds}\}}{\#\{\text{accepted } \mathbf{x}^i\}} \;=\; \frac{c_1}{c_2} \;=\; \frac{Av_n(\breve{\gamma})}{Av_n(\breve{\beta})},$$

the estimate of Section 15.5.

14.3 Exercises 15.15–15.21

Problem 15.15. Consider the bounds on the absolute and relative errors of direct sampling that we provided in Sections 15.4.1 and 15.4.2 (based on Chebyshev's inequality). Derive similar bounds based on this inequality for the absolute and relative errors of (idealized) importance sampling (Theorem 15.12). Can we use Hoeffding's inequality for this purpose? Why?
Recall that Theorem 15.12 treats the idealized case in which the original distribution Pr and the importance distribution $\widetilde{\Pr}$ are proportional at the event α (Definition 15.4), that is, $\Pr(x)/\widetilde{\Pr}(x) = c$ for some constant $c > 0$ and all instantiations x consistent with α. Under that condition the importance sampling function $\tilde{\alpha}$ of Definition 15.3 has variance

$$\mathrm{Va}(\tilde{\alpha}) = \frac{\Pr(\alpha)}{\widetilde{\Pr}(\alpha)} \Pr(\alpha) - \Pr(\alpha)^2.$$

Solution. Corollaries 5 and 7 hold verbatim with Pr replaced by $\widetilde{\Pr}$ in the bound; Hoeffding does not apply in general, since $\tilde{\alpha}$ is not $\{0,1\}$-valued and has no a priori range.
Absolute error. By Theorem 15.11 $\tilde{\alpha}$ has expectation $\Pr(\alpha)$ under $\widetilde{\Pr}$, and in the idealized case of Theorem 15.12 its variance is $\sigma^2 = \Pr(\alpha)^2/\widetilde{\Pr}(\alpha) - \Pr(\alpha)^2 = \Pr(\alpha)^2\widetilde{\Pr}(\neg\alpha)/\widetilde{\Pr}(\alpha)$. So by Theorem 15.5 the estimate (15.8) over a sample drawn from $\widetilde{\Pr}$ has expectation $\Pr(\alpha)$ and variance σ^2/n, and Chebyshev's inequality (Theorem 15.2) applied to it gives for every $\epsilon > 0$

$$P\big(|\mathrm{Av}_n(\tilde{\alpha}) - \Pr(\alpha)| < \epsilon\big) \;\geq\; 1 - \frac{\Pr(\alpha)^2\, \widetilde{\Pr}(\neg\alpha)}{n\, \epsilon^2\, \widetilde{\Pr}(\alpha)}.$$

This is the analogue of Corollary 5, whose bound is $1 - \Pr(\alpha)\Pr(\neg\alpha)/n\epsilon^2$.
Relative error. The relative error is $< \epsilon$ exactly when the absolute error is $< \epsilon \Pr(\alpha)$, so substituting $\epsilon \Pr(\alpha)$ for ϵ above, where the factors $\Pr(\alpha)^2$ cancel,

$$P\left(\frac{|\mathrm{Av}_n(\tilde{\alpha}) - \Pr(\alpha)|}{\Pr(\alpha)} < \epsilon\right) \;\geq\; 1 - \frac{\widetilde{\Pr}(\neg\alpha)}{n\, \epsilon^2\, \widetilde{\Pr}(\alpha)}.$$

This is Corollary 7 with Pr replaced by $\widetilde{\Pr}$ throughout, hence independent of the probability of the event being estimated.

Hoeffding. Not in the form of Theorem 15.8, which requires a $\{0,1\}$-valued function: by Definition 15.3 an importance sampling function takes the value $\Pr(x)/\widetilde{\Pr}(x)$ on instantiations consistent with α, unbounded as $\widetilde{\Pr}(x) \to 0$, while the general form of the inequality needs a known interval $[a,b]$. Under the proportionality of Theorem 15.12 the obstacle disappears: $\tilde{\alpha}(x) = c$ for x consistent with α and 0 otherwise, so $\tilde{\alpha} = c\,\breve{\alpha}$ with $c = \Pr(\alpha)/\widetilde{\Pr}(\alpha)$, summing $\Pr(x) = c\,\widetilde{\Pr}(x)$ over $x \models \alpha$. Theorem 15.8 does apply to the $\{0,1\}$-valued $\breve{\alpha}$, whose expectation under $\widetilde{\Pr}$ is $\widetilde{\Pr}(\alpha)$; putting $\epsilon = c\,\delta$, and then $\epsilon\Pr(\alpha)$ for ϵ,

$$P\left(\frac{|\mathrm{Av}_n(\tilde{\alpha}) - \Pr(\alpha)|}{\Pr(\alpha)} \le \epsilon\right) \;\ge\; 1 - 2e^{-2n\epsilon^2\widetilde{\Pr}(\alpha)^2},$$

which is Corollary 8 with $\Pr(\alpha)$ replaced by $\widetilde{\Pr}(\alpha)$.

Problem 15.16. Consider a simple Bayesian network of the form $A \to B$, where A and B are binary variables. Provide a closed form for the variance of likelihood weighting when estimating $\Pr(a)$. Give a similar form for the estimate of $\Pr(b)$. Your forms must be expressed in terms of network parameters and the probabilities of a and b.
The network has a single edge from A to B; its parameters are θ_a and $\theta_{\bar{a}} = 1 - \theta_a$ for the root A, and $\theta_{b|a}, \theta_{\bar{b}|a} = 1 - \theta_{b|a}, \theta_{b|\bar{a}}, \theta_{\bar{b}|\bar{a}} = 1 - \theta_{b|\bar{a}}$ for B.

Solution. $\mathrm{Va}(\tilde{a}) = 0$, and $\mathrm{Va}(\tilde{b}) = \Pr(a)\Pr(\neg a)(\theta_{b|a} - \theta_{b|\bar{a}})^2$. In each case build the likelihood-weighting network $\widetilde{N}$ of Definition 15.5, read off the importance sampling function, and apply Theorem 15.11:

$$\mathrm{Va}(\tilde{\alpha}) = \left(\sum_{x \models \alpha} \frac{\Pr(x)^2}{\widetilde{\Pr}(x)}\right) - \Pr(\alpha)^2.$$

(i) Estimating $\Pr(a)$. Here $\mathbf{E} = \{A\}$; A is a root, so no edge is deleted and Definition 15.5 merely replaces its CPT by $\tilde{\theta}_a = 1$, $\tilde{\theta}_{\bar{a}} = 0$, leaving the CPT of B untouched, whence $\widetilde{\Pr}(a,b) = \theta_{b|a}$, $\widetilde{\Pr}(a,\bar{b}) = \theta_{\bar{b}|a}$, $\widetilde{\Pr}(\bar{a},\cdot) = 0$. By Theorem 15.14 the weight at either instantiation consistent with a is the single evidence parameter θ_a, so $\tilde{a} \equiv \theta_a = \Pr(a)$ on everything $\widetilde{\Pr}$ can generate and

$$\mathrm{Va}(\tilde{a}) = \theta_a^2\big(\theta_{b|a} + \theta_{\bar{b}|a}\big) - \theta_a^2 = \theta_a^2 - \theta_a^2 = 0.$$

(ii) Estimating $\Pr(b)$. Now $\mathbf{E} = \{B\}$, so Definition 15.5 deletes the edge $A \to B$ and sets $\tilde{\theta}_b = 1$, $\tilde{\theta}_{\bar{b}} = 0$, leaving the CPT of A unchanged, so $\widetilde{\Pr}(a,b) = \theta_a$, $\widetilde{\Pr}(\bar{a},b) = 1 - \theta_a$, and $\widetilde{\Pr}(\cdot,\bar{b}) = 0$. On the two instantiations consistent with b, again by Theorem 15.14, $\tilde{b}(a,b) = \theta_a\theta_{b|a}/\theta_a = \theta_{b|a}$ and $\tilde{b}(\bar{a},b) = \theta_{\bar{a}}\theta_{b|\bar{a}}/\theta_{\bar{a}} = \theta_{b|\bar{a}}$. Theorem 15.11 then gives

$$\begin{aligned}\mathrm{Va}(\tilde{b}) &= \frac{(\theta_a\theta_{b|a})^2}{\theta_a} + \frac{(\theta_{\bar{a}}\theta_{b|\bar{a}})^2}{\theta_{\bar{a}}} - \Pr(b)^2 \\ &= \theta_a\,\theta_{b|a}^2 + (1-\theta_a)\,\theta_{b|\bar{a}}^2 - \Pr(b)^2.\end{aligned}$$

with $\Pr(a) = \theta_a$ and $\Pr(b) = \theta_a\theta_{b|a} + (1-\theta_a)\theta_{b|\bar{a}}$; completing the square gives the equivalent

$$\mathrm{Va}(\tilde{b}) = \Pr(a)\,\Pr(\neg a)\,\big(\theta_{b|a} - \theta_{b|\bar{a}}\big)^2.$$

Problem 15.17. Prove Theorem 15.14.
Theorem 15.14. Let N be a Bayesian network, $\mathbf{e}$ be some evidence, and let $\widetilde{N}$ be the corresponding likelihood-weighting network. Suppose that networks N and $\widetilde{N}$ induce distributions $\Pr(\mathbf{X})$ and $\widetilde{\Pr}(\mathbf{X})$, respectively, and that x is consistent with evidence $\mathbf{e}$. If $\theta_{e|u}$ ranges over the parameters of

variables $E \in \mathbf{E}$ in network N, where eu is consistent with instantiation x, then

$$\Pr(x)/\widetilde{\Pr}(x) = \prod_{\theta_{e|u}} \theta_{e|u} \;\leq\; 1.$$

Recall Definition 15.5: the likelihood-weighting network $\widetilde{N}$ is obtained from N by deleting the edges going into the nodes $\mathbf{E}$ while setting the CPTs of these nodes as follows. If variable $E \in \mathbf{E}$ is instantiated to e in evidence $\mathbf{e}$, then $\tilde{\theta}_e = 1$; otherwise $\tilde{\theta}_e = 0$. All other CPTs of $\widetilde{N}$ are equal to those in N.

Solution. Factor both distributions by the chain rule (4.2) and cancel; it applies to $\widetilde{N}$ since deleting edges creates no cycle and each $E \in \mathbf{E}$ becomes a root whose new CPT is a legitimate distribution over E. Write x_X for the value x assigns X and u_X for what it assigns the parents of X in N. Every $X \notin \mathbf{E}$ keeps its parents and CPT in $\widetilde{N}$, and for each $E \in \mathbf{E}$ the value x_E is the value e that $\mathbf{e}$ assigns E (x being consistent with $\mathbf{e}$), so $\tilde{\theta}_{x_E} = \tilde{\theta}_e = 1$ by Definition 15.5 and

$$\widetilde{\Pr}(x) = \prod_{X \notin \mathbf{E}} \theta_{x_X|u_X}.$$

Splitting the chain rule for $\Pr(x)$ along the same partition of $\mathbf{X}$ gives

$$\Pr(x) = \left(\prod_{X \notin \mathbf{E}} \theta_{x_X|u_X} \right) \cdot \prod_{E \in \mathbf{E}} \theta_{x_E|u_E} = \widetilde{\Pr}(x) \cdot \prod_{E \in \mathbf{E}} \theta_{x_E|u_E}.$$

Each $\theta_{x_E|u_E}$ in the second product is a parameter of an evidence variable of N whose family instantiation $x_E u_E$ is consistent with x — exactly the $\theta_{e|u}$ of the statement. Dividing wherever $\widetilde{\Pr}(x) \neq 0$,

$$\frac{\Pr(x)}{\widetilde{\Pr}(x)} = \prod_{\theta_{e|u}} \theta_{e|u} \;\leq\; 1,$$

the bound because a finite product of parameters $0 \leq \theta_{e|u} \leq 1$ lies in $[0, 1]$. The undivided form holds for every x consistent with $\mathbf{e}$, so $\widetilde{\Pr}(x) = 0$ forces $\Pr(x) = 0$.

Problem 15.18. Consider a Bayesian network in which all leaf nodes are deterministic (their CPTs contain only 0 and 1 entries) and let $\mathbf{e}$ be an instantiation of the leaf nodes. What is the variance of likelihood weighting when estimating $\Pr(\mathbf{e})$?

Solution. Likelihood weighting buys nothing here: its variance is

$$\mathrm{Va}(\tilde{\mathbf{e}}) = \Pr(\mathbf{e}) \Pr(\neg \mathbf{e}) = \Pr(\mathbf{e}) - \Pr(\mathbf{e})^2,$$

exactly the direct sampling variance of Theorem 15.4, the worst case allowed by Exercise 15.19.
The weights are 0/1-valued. With $\mathbf{E}$ the leaf nodes and $\widetilde{N}$ the likelihood-weighting network of Definition 15.5 inducing $\widetilde{\Pr}$, Theorem 15.14 gives for x consistent with $\mathbf{e}$

$$w(x) \stackrel{\text{def}}{=} \frac{\Pr(x)}{\widetilde{\Pr}(x)} = \prod_{E \in \mathbf{E}} \theta_{x_E|u_E},$$

one parameter per leaf, the one whose family instantiation is consistent with x. Every CPT entry of a leaf is 0 or 1 by hypothesis, so $w(x) \in \{0, 1\}$ for every $x \models \mathbf{e}$. By Theorem 15.11 the importance sampling function $\tilde{\mathbf{e}}$ has variance

$$\mathrm{Va}(\tilde{\mathbf{e}}) = \left(\sum_{x \models \mathbf{e}} \frac{\Pr(x)^2}{\widetilde{\Pr}(x)} \right) - \Pr(\mathbf{e})^2.$$

For $x \models \mathbf{e}$ with $\widetilde{\Pr}(x) > 0$ we have $\Pr(x) = w(x)\widetilde{\Pr}(x)$ and $w(x)^2 = w(x)$, so the summand is $w(x)^2\widetilde{\Pr}(x) = w(x)\widetilde{\Pr}(x) = \Pr(x)$; if instead $\widetilde{\Pr}(x) = 0$ then $\Pr(x) = 0$ (Exercise 15.17) and the term again contributes $\Pr(x)$. Hence the sum is $\sum_{x\models\mathbf{e}} \Pr(x) = \Pr(\mathbf{e})$, giving $\mathrm{Va}(\tilde{\mathbf{e}}) = \Pr(\mathbf{e}) - \Pr(\mathbf{e})^2$.

Problem 15.19. Consider the functions $\breve{\alpha}$ and $\tilde{\alpha}$ given in Definitions 15.1 and 15.3, respectively. Show that $\mathrm{Va}(\tilde{\alpha}) \leq \mathrm{Va}(\breve{\alpha})$ if $\widetilde{\Pr}(x) \geq \Pr(x)$ for all instantiations x consistent with α.
Here $\breve{\alpha}(x) = 1$ if α is true at x and 0 otherwise (Definition 15.1), while $\tilde{\alpha}(x) = \Pr(x)/\widetilde{\Pr}(x)$ if α is true at x and 0 otherwise (Definition 15.3), where the importance distribution $\widetilde{\Pr}$ is required to satisfy $\widetilde{\Pr}(x) = 0$ only if $\Pr(x) = 0$, for all x at which α is true. The variance $\mathrm{Va}(\breve{\alpha})$ is taken with respect to Pr and $\mathrm{Va}(\tilde{\alpha})$ with respect to $\widetilde{\Pr}$.

Solution. The two variances share the term $-\Pr(\alpha)^2$: by Theorem 15.4 $\mathrm{Va}(\breve{\alpha}) = \Pr(\alpha) - \Pr(\alpha)^2$, while by Theorem 15.11

$$\mathrm{Va}(\tilde{\alpha}) = \left(\sum_{x\models\alpha} \frac{\Pr(x)^2}{\widetilde{\Pr}(x)}\right) - \Pr(\alpha)^2,$$

so it suffices to bound each term of that sum by $\Pr(x)$. Fix $x \models \alpha$. (i) If $\widetilde{\Pr}(x) = 0$, then the requirement in Definition 15.3 forces $\Pr(x) = 0$, $\tilde{\alpha}(x)$ is taken to be 0, and the term is $0 = \Pr(x)$. (ii) If $\widetilde{\Pr}(x) > 0$ and $\Pr(x) = 0$, the term is again $0 = \Pr(x)$; and if $\Pr(x) > 0$, the hypothesis $\widetilde{\Pr}(x) \geq \Pr(x)$ enlarges the positive denominator, so

$$\frac{\Pr(x)^2}{\widetilde{\Pr}(x)} \;\leq\; \frac{\Pr(x)^2}{\Pr(x)} \;=\; \Pr(x).$$

Summing over $x \models \alpha$ gives $\sum_{x\models\alpha} \Pr(x)^2/\widetilde{\Pr}(x) \leq \Pr(\alpha)$, and subtracting $\Pr(\alpha)^2$,

$$\mathrm{Va}(\tilde{\alpha}) \;\leq\; \Pr(\alpha) - \Pr(\alpha)^2 \;=\; \mathrm{Va}(\breve{\alpha}).$$

Problem 15.20. Show the following: If $\Pr(.|\alpha)$ can be used as an importance distribution, then $\Pr(\alpha)$ can be computed exactly after generating a sample of size 1.

Solution. The importance sampling function is then constant at $\Pr(\alpha)$, so a single draw returns it exactly. Assume $\Pr(\alpha) > 0$, as otherwise $\Pr(.|\alpha)$ is undefined, and put

$$\widetilde{\Pr}(x) \;=\; \Pr(x \mid \alpha) \;=\; \begin{cases} \dfrac{\Pr(x)}{\Pr(\alpha)}, & \text{if } \alpha \text{ is true at } x \\ 0, & \text{otherwise.} \end{cases}$$

This is legal: at an x satisfying α, $\widetilde{\Pr}(x) = \Pr(x)/\Pr(\alpha)$ vanishes iff $\Pr(x)$ does, as Definition 15.3 demands. Any x with $\widetilde{\Pr}(x) > 0$ satisfies α, so by Definition 15.3

$$\tilde{\alpha}(x) \;=\; \frac{\Pr(x)}{\widetilde{\Pr}(x)} \;=\; \frac{\Pr(x)}{\Pr(x)/\Pr(\alpha)} \;=\; \Pr(\alpha)$$

on the whole support of $\widetilde{\Pr}$. Sampling produces only instantiations of positive probability, so $\mathrm{Av}_1(\tilde{\alpha}) = \tilde{\alpha}(x^1) = \Pr(\alpha)$ for every possible outcome x^1.

Problem 15.21. Consider a Bayesian network $X_1 \to \ldots \to X_n$ (a chain in which each X_i for $i > 1$ has the single parent X_{i-1}, and X_1 is a root) and suppose that we want to estimate $\Pr(x_n)$ using likelihood weighting. Find a closed form for the variance of this method in terms of the CPT for

node X_n and the marginal distribution of node X_{n-1}. Identify conditions on the network that lead to a zero variance.

Solution. $\mathrm{Va}(\tilde{\alpha}) = \sum_{x_{n-1}} \Pr(x_{n-1})(\theta_{x_n|x_{n-1}} - \Pr(x_n))^2$, zero exactly when the CPT column of X_n at x_n is constant over the support of $\Pr(X_{n-1})$. The evidence is $\mathbf{e} = x_n$ on $\mathbf{E} = \{X_n\}$, so Definition 15.5 deletes the edge $X_{n-1} \to X_n$ and makes the CPT of X_n degenerate at x_n, leaving the structure $X_1 \to \ldots \to X_{n-1}$ and its parameters verbatim. Hence

$$\widetilde{\Pr}(x_1, \ldots, x_{n-1}, x_n) = \Pr(x_1, \ldots, x_{n-1}),$$

zero at any other value of X_n; in particular X_{n-1} keeps its true marginal $\Pr(X_{n-1})$. With X_n the sole evidence variable, whose parent takes value x_{n-1} in x, Theorem 15.14 gives

$$\tilde{\alpha}(x) = \frac{\Pr(x)}{\widetilde{\Pr}(x)} = \theta_{x_n|x_{n-1}},$$

read straight off the CPT of X_n. So Theorem 15.11, with $\Pr(x) = \Pr(x_1, \ldots, x_{n-1})\theta_{x_n|x_{n-1}}$, gives $\mathrm{Va}(\tilde{\alpha}) = \sum_{x_1,\ldots,x_{n-1}} \Pr(x_1, \ldots, x_{n-1})\theta^2_{x_n|x_{n-1}} - \Pr(x_n)^2$, whose summand depends on $x_1, \ldots, x_{n-2}$ only through the leading factor; summing those out gives the required closed form

$$\boxed{\mathrm{Va}(\tilde{\alpha}) = \left(\sum_{x_{n-1}} \Pr(x_{n-1})\, \theta^2_{x_n|x_{n-1}} \right) - \Pr(x_n)^2}$$

with $\Pr(x_n) = \sum_{x_{n-1}} \Pr(x_{n-1})\theta_{x_n|x_{n-1}}$: only the CPT column of X_n at x_n and the marginal $\Pr(X_{n-1})$ enter, as promised. Completing the square gives the equivalent form

$$\mathrm{Va}(\tilde{\alpha}) = \sum_{x_{n-1}} \Pr(x_{n-1})\big(\theta_{x_n|x_{n-1}} - \Pr(x_n)\big)^2.$$

A weighted sum of squares vanishes only if each term does, so $\mathrm{Va}(\tilde{\alpha}) = 0$ iff $\theta_{x_n|x_{n-1}} = \Pr(x_n)$ for every x_{n-1} with $\Pr(x_{n-1}) > 0$. Two conditions on the network realise this:

- (i) The CPT rows of X_n are identical, $\theta_{x_n|x_{n-1}} = \theta_{x_n}$ for all x_{n-1}; then by the chain rule $\Pr(x_n \mid x_1, \ldots, x_{n-1}) = \theta_{x_n}$ at every prefix of positive probability, so every sample returns the weight $\theta_{x_n} = \Pr(x_n)$. (A parameter property, not a structural one: X_{n-1} is a parent of X_n, so d-separation certifies nothing here.)
- (ii) $\Pr(X_{n-1})$ is degenerate, $\Pr(x^*_{n-1}) = 1$, so the condition need hold at one value only — as when $\theta_{x_1} = 1$ for one value of X_1 and every CPT $\theta_{X_i|X_{i-1}}$, $2 \le i \le n-1$, is 0/1-valued.

14.4 Exercises 15.22–15.28

Problem 15.22. Consider the HMM in Figure 15.14 and suppose that our goal is to estimate the probability $\Pr(S_t \mid o_1, \ldots, o_t)$ for each time t given evidence $o_1, \ldots, o_t$. Suppose that we use likelihood weighting for this purpose and let $\widehat{\Pr}(.)$ be the importance distribution. Show that for every instantiation $s_1, \ldots, s_t, o_1, \ldots, o_t$ that satisfies the given evidence, we have

$$\frac{\Pr(s_1, \ldots, s_t, o_1, \ldots, o_t)}{\widehat{\Pr}(s_1, \ldots, s_t, o_1, \ldots, o_t)} = \Pr(o_1, \ldots, o_t \mid s_1, \ldots, s_t).$$

Describe an implementation of this algorithm whose space complexity is independent of time t.
Figure 15.14 shows, in part (a), an HMM for three time steps: a state chain $S_1 \to S_2 \to S_3$ together with a sensor edge $S_i \to O_i$ for each $i = 1, 2, 3$. Part (b) shows the corresponding likelihood-weighting HMM given evidence on nodes O_1, O_2, O_3: it has the same nodes and the same state edges $S_1 \to S_2 \to S_3$, but every sensor edge $S_i \to O_i$ has been deleted, so each O_i is a root whose CPT assigns probability 1 to its observed value o_i and 0 to all other values (Definition 15.5). The

general t-step HMM and its likelihood-weighting network are obtained in the same way: the HMM is specified by an initial distribution $\Pr(S_1)$, a transition distribution $\Pr(S_t \mid S_{t-1})$, and a sensor distribution $\Pr(O_t \mid S_t)$.

Solution. Both sides equal $\prod_{i=1}^{t} \Pr(o_i \mid s_i)$. The chain rule (4.2) in the HMM N gives

$$\Pr(s_1, \dots, s_t, o_1, \dots, o_t) = \Pr(s_1) \prod_{i=2}^{t} \Pr(s_i \mid s_{i-1}) \cdot \prod_{i=1}^{t} \Pr(o_i \mid s_i),$$

while in the likelihood-weighting network $\widehat{\mathsf{N}}$ of Definition 15.5 the edges $S_i \to O_i$ are deleted and each O_i carries $\widehat{\theta}_{o_i} = 1$ on the observed value, the state CPTs being untouched, so at an instantiation satisfying the evidence the chain rule gives

$$\widehat{\Pr}(s_1, \dots, s_t, o_1, \dots, o_t) = \Pr(s_1) \prod_{i=2}^{t} \Pr(s_i \mid s_{i-1}),$$

the prior over the state trajectory — positive whenever the trajectory itself is, so Definition 15.3 is satisfied. Dividing gives the ratio $\prod_{i=1}^{t} \Pr(o_i \mid s_i)$, which is Theorem 15.14 for this network (the evidence parameters consistent with the instantiation are the $\theta_{o_i|s_i}$). Summing the O_i, which are leaves, out of the first display leaves $\Pr(s_1, \dots, s_t) = \Pr(s_1) \prod_{i=2}^{t} \Pr(s_i \mid s_{i-1})$, and dividing the joint by this marginal identifies that same product with $\Pr(o_1, \dots, o_t \mid s_1, \dots, s_t)$, as required.
For the implementation write $w(s_1, \dots, s_t) = \prod_{i=1}^{t} \Pr(o_i \mid s_i)$. By Definition 15.3 and Theorem 15.11 likelihood weighting estimates an event's probability by the sample mean of its importance sampling function over n draws from $\widehat{\Pr}$; taking $\beta = o_1, \dots, o_t$ and $\gamma = s_t \wedge \beta$, and noting every sample satisfies β, the two sample means are $\frac{1}{n}\sum_i w^i$ and $\frac{1}{n}\sum_{i : s_t^i = s_t} w^i$ with $w^i = w(s_1^i, \dots, s_t^i)$, so

$$\Pr(s_t \mid o_1, \dots, o_t) \approx \frac{\sum_{i \,:\, s_t^i = s_t} w^i}{\sum_{i=1}^{n} w^i}.$$

Both the sampling and the weight are recursive in t: sampling from $\widehat{\Pr}$ visits nodes in topological order (Algorithm 44), so s_t^i is drawn from $\Pr(S_t \mid s_{t-1}^i)$ and depends on the past only through s_{t-1}^i, while $w_t^i = w_{t-1}^i \Pr(o_t \mid s_t^i)$ with $w_0^i = 1$. So no trajectory need be stored — only the pair (s_t^i, w_t^i) per sample, observations processed one at a time:

1. Initialize: for $i = 1, \dots, n$, sample s_1^i from $\Pr(S_1)$ and set $w_1^i \leftarrow \Pr(o_1 \mid s_1^i)$.
2. For each $t > 1$ and each i: sample s_t^i from $\Pr(S_t \mid s_{t-1}^i)$, overwriting s_{t-1}^i; then set $w_t^i \leftarrow w_{t-1}^i \Pr(o_t \mid s_t^i)$, overwriting w_{t-1}^i.
3. To report the estimate at time t, accumulate $P \leftarrow \sum_i w_t^i$ and $P[s] \leftarrow \sum_{i : s_t^i = s} w_t^i$, and return $P[s]/P$ for each state s.

Storage is n states plus n weights plus a table of size $|S|$ (or an $O(n)$ hash table keyed by the states actually sampled), i.e. $O(n + |S|)$, independent of t; the work per time step is $O(n)$. Keep the w_t^i in log-space or rescale them by their sum each step, since the common factor cancels in the ratio above.

Problem 15.23. Show that Equations 15.11 and 15.12 are implied by Equation 15.10.
Context. Particle filtering on the HMM of Figure 15.8 (a state chain $S_1 \to S_2 \to \cdots \to S_n$ with a sensor edge $S_t \to O_t$ at each time) replaces the full HMM by a sequence of three-node network fragments. For each time $t > 1$ the fragment of Figure 15.9 induces a distribution $\Pr'(S_{t-1}, S_t, O_t)$ and is specified by three CPTs: a CPT $\Pr^\star(S_{t-1})$ for the root S_{t-1}, the transition CPT $\Pr(S_t \mid S_{t-1})$ taken from the HMM, and the sensor CPT $\Pr(O_t \mid S_t)$ taken from the HMM. Its edges are $S_{t-1} \to S_t$ and $S_t \to O_t$, so

$$\Pr'(S_{t-1}, S_t, O_t) = \Pr^\star(S_{t-1}) \Pr(S_t \mid S_{t-1}) \Pr(O_t \mid S_t).$$

Equation 15.10 is the choice

$$\overset{\star}{\Pr}(S_{t-1}) = \Pr(S_{t-1} \mid O_1, \ldots, O_{t-1}),$$

where $O_1, \ldots, O_{t-1}$ are fixed at their observed values. Equations 15.11 and 15.12 are

$$\overset{\prime}{\Pr}(S_{t-1}, S_t, O_t) = \Pr(S_{t-1}, S_t, O_t \mid O_1, \ldots, O_{t-1}),$$

$$\overset{\prime}{\Pr}(S_t \mid O_t) = \Pr(S_t \mid O_1, \ldots, O_t).$$

Solution. Equation 15.11 is the chain rule under $\Pr(\cdot \mid \mathbf{o}_{<t})$ with the two Markov properties of the HMM substituted, and Equation 15.12 follows from it by summing out. Write $\mathbf{o}_{<t}$ for $O_1 = o_1, \ldots, O_{t-1} = o_{t-1}$, assumed of positive probability. In the order S_{t-1}, S_t, O_t,

$$\begin{aligned}&\Pr(s_{t-1}, s_t, o_t \mid \mathbf{o}_{<t})\\ &\quad = \Pr(s_{t-1} \mid \mathbf{o}_{<t}) \cdot \Pr(s_t \mid s_{t-1}, \mathbf{o}_{<t}) \cdot \Pr(o_t \mid s_t, s_{t-1}, \mathbf{o}_{<t}).\end{aligned}$$

The last two factors simplify by d-separation (Definition 4.1, Theorem 4.1). Any path from S_t to an O_i, $i < t$, must leave S_t backwards through S_{t-1}, a sequential valve that is instantiated (forward it reaches only the leaf O_t or nodes with index $> t$), so S_t is d-separated from $O_1, \ldots, O_{t-1}$ by S_{t-1}; and O_t's only incident edge is $S_t \to O_t$, so every path out of it passes through the instantiated sequential valve S_t and O_t is d-separated from $S_{t-1}, O_1, \ldots, O_{t-1}$ by S_t. Hence $\Pr(s_t \mid s_{t-1}, \mathbf{o}_{<t}) = \Pr(s_t \mid s_{t-1})$ and $\Pr(o_t \mid s_t, s_{t-1}, \mathbf{o}_{<t}) = \Pr(o_t \mid s_t)$; substituting these and then Equation 15.10,

$$\begin{aligned}\Pr(s_{t-1}, s_t, o_t \mid \mathbf{o}_{<t}) &= \Pr(s_{t-1} \mid \mathbf{o}_{<t}) \Pr(s_t \mid s_{t-1}) \Pr(o_t \mid s_t)\\ &= \overset{\star}{\Pr}(s_{t-1}) \Pr(s_t \mid s_{t-1}) \Pr(o_t \mid s_t)\\ &= \overset{\prime}{\Pr}(s_{t-1}, s_t, o_t),\end{aligned}$$

the last step being the chain rule for the fragment of Figure 15.9. Holding for every s_{t-1}, s_t, o_t, this is Equation 15.11. Summing it over S_{t-1} gives $\Pr'(s_t, o_t) = \Pr(s_t, o_t \mid \mathbf{o}_{<t})$, and summing further over S_t gives $\Pr'(o_t) = \Pr(o_t \mid \mathbf{o}_{<t})$, assumed positive (equivalently $\Pr(o_1, \ldots, o_t) > 0$). Then

$$\begin{aligned}\overset{\prime}{\Pr}(s_t \mid o_t) = \frac{\Pr'(s_t, o_t)}{\Pr'(o_t)} &= \frac{\Pr(s_t, o_t \mid \mathbf{o}_{<t})}{\Pr(o_t \mid \mathbf{o}_{<t})}\\ &= \frac{\Pr(s_t, o_t, \mathbf{o}_{<t}) / \Pr(\mathbf{o}_{<t})}{\Pr(o_t, \mathbf{o}_{<t}) / \Pr(\mathbf{o}_{<t})} = \frac{\Pr(s_t, o_1, \ldots, o_t)}{\Pr(o_1, \ldots, o_t)}\\ &= \Pr(s_t \mid o_1, \ldots, o_t),\end{aligned}$$

which is Equation 15.12.

Problem 15.24. Prove Equations 15.13 and 15.14. That is, show that these are the estimates computed by likelihood weighting when applied to the network fragment at time $t > 1$.
Context. At time $t > 1$ particle filtering works with the three-node fragment of Figure 15.9, with edges $S_{t-1} \to S_t$ and $S_t \to O_t$, whose CPTs are $\Pr^{\star}(S_{t-1})$, $\Pr(S_t \mid S_{t-1})$, and $\Pr(O_t \mid S_t)$; it induces

$$\overset{\prime}{\Pr}(S_{t-1}, S_t, O_t) = \overset{\star}{\Pr}(S_{t-1}) \Pr(S_t \mid S_{t-1}) \Pr(O_t \mid S_t).$$

The corresponding likelihood-weighting network, Figure 15.10(b), is obtained by Definition 15.5: the edge $S_t \to O_t$ is deleted and the CPT of the root O_t is set to $\widehat{\theta}_{o_t} = 1$ on the observed value o_t and 0 elsewhere. Its distribution is written $\widehat{\Pr}$. Likelihood weighting generates from this network the sample

$$s_{t-1}^1, s_t^1, o_t; \quad s_{t-1}^2, s_t^2, o_t; \quad \ldots; \quad s_{t-1}^n, s_t^n, o_t,$$

and estimates $\Pr'(s_t \mid o_t)$ as the ratio of estimates of $\Pr'(s_t, o_t)$ and $\Pr'(o_t)$. Show that the estimate of $\Pr'(o_t)$ is given by Equation 15.13, namely

$$\frac{1}{n}\sum_{i=1}^{n} \Pr(o_t \mid s_t^i)$$

and that the estimate of $\Pr'(s_t, o_t)$ is given by Equation 15.14, namely

$$\frac{1}{n}\sum_{\substack{i \\ s_t^i = s_t}} \Pr(o_t \mid s_t^i),$$

so that the estimate of $\Pr'(s_t \mid o_t)$ is given by Equation 15.15, namely

$$\frac{\sum_{i\,:\,s_t^i = s_t} \Pr(o_t \mid s_t^i)}{\sum_i \Pr(o_t \mid s_t^i)}.$$

Solution. Every sample carries the weight $\Pr(o_t \mid s_t)$, and Equations 15.13–15.15 are the sample means that follow. The likelihood-weighting network of Figure 15.10(b) has the single edge $S_{t-1} \to S_t$ and O_t a root, so its chain rule gives

$$\widehat{\Pr}(s_{t-1}, s_t, o) = \overset{\star}{\Pr}(s_{t-1}) \Pr(s_t \mid s_{t-1}) \widehat{\theta}_o,$$

with $\widehat{\theta}_o = 1$ iff $o = o_t$. Hence all the mass sits on instantiations satisfying $O_t = o_t$, where $\widehat{\Pr}(s_{t-1}, s_t, o_t) = \Pr^\star(s_{t-1}) \Pr(s_t \mid s_{t-1})$: sampling by Algorithm 44 draws s_{t-1}^i from $\Pr^\star(S_{t-1})$, then s_t^i from $\Pr(S_t \mid s_{t-1}^i)$, and sets $O_t = o_t$, which is the sample displayed in the statement. And for every $\mathbf{x} = s_{t-1}, s_t, o_t$ consistent with the evidence,

$$\frac{\Pr'(\mathbf{x})}{\widehat{\Pr}(\mathbf{x})} = \frac{\Pr^\star(s_{t-1}) \Pr(s_t \mid s_{t-1}) \Pr(o_t \mid s_t)}{\Pr^\star(s_{t-1}) \Pr(s_t \mid s_{t-1})} = \Pr(o_t \mid s_t).$$

This is Theorem 15.14 here, O_t being the sole evidence variable and contributing the single parameter $\theta_{o_t \mid s_t} = \Pr(o_t \mid s_t)$.

For Equation 15.13 take $\beta = (O_t = o_t)$, with importance sampling function $\widetilde{\beta} = \Pr'/\widehat{\Pr}$ on $\mathbf{x} \models \beta$ and 0 elsewhere (Definition 15.3, admissible here since $\widehat{\Pr}(\mathbf{x}) = 0$ at an $\mathbf{x} \models \beta$ forces $\Pr^\star(s_{t-1}) \Pr(s_t \mid s_{t-1}) = 0$ and hence $\Pr'(\mathbf{x}) = 0$). By Theorem 15.11 $\mathrm{Ex}(\widetilde{\beta}) = \Pr'(o_t)$, and every sampled $\mathbf{x}^i$ satisfies β, so by the ratio above $\widetilde{\beta}(\mathbf{x}^i) = \Pr(o_t \mid s_t^i)$ and $\mathrm{Av}_n(\widetilde{\beta}) = \frac{1}{n}\sum_i \Pr(o_t \mid s_t^i)$, Equation 15.13, accumulated on line 5 of Algorithm 47.

For Equation 15.14 fix s_t and take $\gamma = (S_t = s_t) \wedge (O_t = o_t)$, whose importance sampling function is $\Pr(o_t \mid s_t)$ on $\mathbf{x} \models \gamma$ and 0 elsewhere, again by that ratio, with $\mathrm{Ex}(\widetilde{\gamma}) = \Pr'(s_t, o_t)$ (Theorem 15.11). Since each sample satisfies $O_t = o_t$ automatically, $\mathbf{x}^i \models \gamma$ exactly when $s_t^i = s_t$, so

$$\mathrm{Av}_n(\widetilde{\gamma}) = \frac{1}{n}\sum_{\substack{i \\ s_t^i = s_t}} \Pr(o_t \mid s_t^i),$$

which is Equation 15.14, line 6 of Algorithm 47. Following Section 15.5, the conditional probability is the ratio of the two estimates, the factors $1/n$ cancelling:

$$\overset{\prime}{\Pr}(s_t \mid o_t) \approx \frac{\mathrm{Av}_n(\widetilde{\gamma})}{\mathrm{Av}_n(\widetilde{\beta})} = \frac{\sum_{i\,:\,s_t^i = s_t} \Pr(o_t \mid s_t^i)}{\sum_i \Pr(o_t \mid s_t^i)},$$

which is Equation 15.15 and the value returned on line 8 of Algorithm 47. Nothing above used any particular form of $\Pr^\star(S_{t-1})$, which in the algorithm is the empirical distribution carried by the n particles from time $t-1$, so drawing s_{t-1}^i from it is just taking the i-th incoming particle — why Algorithm 47 has no explicit sampling step for S_{t-1}.

Problem 15.25. Write the pseudocode for applying particle filtering to the class of DBNs discussed in Section 15.6.2 and Figure 15.12.

Context. The class of DBNs in question (Figure 15.11) satisfies two conditions: every edge either connects two nodes at the same time t or goes from a node at time $t-1$ to a node at time t; and the nodes at time t split into two sets $\mathbf{S}_t$ and $\mathbf{O}_t$, where the nodes $\mathbf{O}_t$ are leaves whose parents lie in $\mathbf{S}_t$ and which are guaranteed to be observed at time t. Such a network is a factored HMM: the hidden state is the set $\mathbf{S}_t$ and the observation is the set $\mathbf{O}_t$.

The concrete DBN of Figure 15.11 has, in each time slice, hidden variables S^1, S^2, S^3, S^4 and observed variables O^1, O^2, with intra-slice edges

$$S_t^1 \to S_t^3, \quad S_t^2 \to S_t^3, \quad S_t^2 \to S_t^4, \quad S_t^3 \to O_t^1, \quad S_t^4 \to O_t^1, \quad S_t^4 \to O_t^2,$$

and inter-slice edges $S_{t-1}^2 \to S_t^2$ and $S_{t-1}^3 \to S_t^3$.

Figure 15.12 shows the two fragments used by particle filtering on this network. Fragment (a) represents $\Pr(\mathbf{S}_t \mid \mathbf{S}_{t-1})$: it consists of the slice at time t restricted to $\mathbf{S}_t$, that is, the nodes $S_t^1, S_t^2, S_t^3, S_t^4$ with the edges $S_t^1 \to S_t^3$, $S_t^2 \to S_t^3$, $S_t^2 \to S_t^4$ among them, together with the two instantiated parents s_{t-1}^2 and s_{t-1}^3 from time $t-1$ and their edges into S_t^2 and S_t^3 respectively. It therefore induces $\Pr(S_t^1, S_t^2, S_t^3, S_t^4 \mid s_{t-1}^2, s_{t-1}^3)$. Fragment (b) represents $\Pr(\mathbf{O}_t \mid \mathbf{S}_t)$: it keeps only the nodes in $\mathbf{O}_t$ and their parents in $\mathbf{S}_t$, here the instantiated nodes s_t^3, s_t^4 with the edges $s_t^3 \to O_t^1$, $s_t^4 \to O_t^1$ and $s_t^4 \to O_t^2$, so that $\Pr(O_t^1, O_t^2 \mid s_t^3, s_t^4) = \Pr(O_t^1 \mid s_t^3, s_t^4)\Pr(O_t^2 \mid s_t^4)$ is a product of network parameters.

Solution. A particle is an instantiation of the interface $\mathbf{I}_{t-1} \subseteq \mathbf{S}_{t-1}$, the hidden variables at time $t-1$ having a child at time t ($\{S^2, S^3\}$ in Figure 15.12(a)); this is all that crosses a time boundary, since no other time-$(t-1)$ node has a child at time t, so given $\mathbf{i}_{t-1}$ the slice at time t is independent of the rest of the past. Two fragments are used, both ordinary Bayesian networks over one slice:

- N^{tr}, the fragment of Figure 15.12(a). Its roots are the instantiated interface nodes from time $t-1$; the remaining nodes are $\mathbf{S}_t$ with the intra-slice edges. With the roots clamped to a particle $\mathbf{i}_{t-1}$ it induces $\Pr(\mathbf{S}_t \mid \mathbf{i}_{t-1})$.
- N^{sn}, the fragment of Figure 15.12(b), used only to read off parameters: for a full state $\mathbf{s}_t$ and the observation $\mathbf{o}_t$,
$$\Pr(\mathbf{o}_t \mid \mathbf{s}_t) = \prod_{O \in \mathbf{O}_t} \theta_{o|u},$$
where o is the value of O in $\mathbf{o}_t$ and u the instantiation of O's parents in $\mathbf{s}_t$ — a product of network parameters precisely because the nodes $\mathbf{O}_t$ are leaves with all parents in $\mathbf{S}_t$, so $O(|\mathbf{O}_t|)$ table lookups and no inference.

One iteration of the filter:

DBN PARTICLE FILTER($\{\mathbf{i}_{t-1}^1, \ldots, \mathbf{i}_{t-1}^n\}$, $\mathbf{o}_t$, N^{tr}, N^{sn})

input: n particles $\mathbf{i}_{t-1}^k$ over $\mathbf{I}_{t-1}$, the evidence $\mathbf{o}_t$, and the two fragments. output: a weighted estimate of $\Pr'(\mathbf{S}_t \mid \mathbf{o}_t)$ and n particles for time $t+1$.

line	statement
1	$P \leftarrow 0$ (estimate of $\Pr'(\mathbf{o}_t)$)
2	$L \leftarrow$ empty hash table mapping states to weights
3	for $k = 1$ to n do
4	clamp the roots of N^{tr} to $\mathbf{i}_{t-1}^k$
5	$\mathbf{s}_t^k \leftarrow$ SIMULATE BN(N^{tr})
6	$w_k \leftarrow \prod_{O \in \mathbf{O}_t} \theta_{o|u}$
7	$P \leftarrow P + w_k$
8	$L[\mathbf{s}_t^k] \leftarrow L[\mathbf{s}_t^k] + w_k$
9	end for
10	if $P = 0$ then report filter failure and stop
11	$L[\mathbf{s}] \leftarrow L[\mathbf{s}]/P$ for each key $\mathbf{s}$ of L
12	draw $\mathbf{s}^1, \ldots, \mathbf{s}^n$ independently from L
13	return L and the n particles $\mathbf{i}_t^k$

Here line 5 is Algorithm 44 on the clamped fragment, line 6 reads its parameters from N^{sn} at u the value of O's parents in $\mathbf{s}_t^k$, line 8 inserts an absent key with weight 0, and $\mathbf{i}_t^k$ on line 13 is the value $\mathbf{I}_t$ takes in $\mathbf{s}^k$. At $t = 1$ omit line 4 and let line 5 be SIMULATE BN on the initial slice restricted to $\mathbf{S}_1$, which induces $\Pr(\mathbf{S}_1)$.
Correctness: lines 4 and 5 sample $\mathbf{s}_t^k$ from $\Pr(\mathbf{S}_t \mid \mathbf{i}_{t-1}^k)$, the vector analogue of line 4 of Algorithm 47, and line 6 computes $\Pr(\mathbf{o}_t \mid \mathbf{s}_t^k)$, which by Exercise 15.24 — whose argument goes through verbatim with S_{t-1}, S_t, O_t replaced by the sets $\mathbf{S}_{t-1}, \mathbf{S}_t, \mathbf{O}_t$ — is the ratio $\Pr'(\mathbf{x})/\widehat{\Pr}(\mathbf{x})$ for the likelihood-weighting network of the time-t fragment. Lines 7, 8, and 11 therefore compute

$$L[\mathbf{s}_t] = \frac{\sum_{k\,:\,\mathbf{s}_t^k = \mathbf{s}_t} \Pr(\mathbf{o}_t \mid \mathbf{s}_t^k)}{\sum_k \Pr(\mathbf{o}_t \mid \mathbf{s}_t^k)},$$

which is Equation 15.15 for the factored case, an estimate of $\Pr'(\mathbf{s}_t \mid \mathbf{o}_t) = \Pr(\mathbf{s}_t \mid \mathbf{o}_1, \ldots, \mathbf{o}_t)$ by Equation 15.12; line 12 draws the next step's particles from it.
L must not be indexed by all instantiations of $\mathbf{S}_t$, whose number m is exponential in $|\mathbf{S}_t|$; keyed by a hash table on the states actually sampled it holds at most n keys, so one iteration runs in $O(nc)$ time and space with $c = |\mathbf{S}_t| + |\mathbf{O}_t|$ the slice size (line 5 costs $O(|\mathbf{S}_t|)$ samplings, line 6 $O(|\mathbf{O}_t|)$ lookups) — the $O(n)$ rather than $O(m)$ behaviour of footnote 6, independent of t.

Problem 15.26. Show that if a Markov chain satisfies the detailed balance property given in Equation 15.16, then $\Pr(\mathbf{X})$ must be a stationary distribution for the chain.
Context. A Markov chain for variables $\mathbf{X}$ is a dynamic Bayesian network $\mathbf{X}_1 \to \mathbf{X}_2 \to \cdots \to \mathbf{X}_n$ whose instantiations $\mathbf{x}$ are called states; the CPT for $\mathbf{X}_1$ is the initial distribution and the CPTs for $\mathbf{X}_2, \ldots, \mathbf{X}_n$ are the transition matrices, all equal since the chain is homogeneous. $\mathbb{P}$ denotes the distribution induced by the chain. The distribution $\Pr(\mathbf{X})$ is a stationary distribution for the chain if setting $\mathbb{P}(\mathbf{X}_1) = \Pr(\mathbf{X})$ yields $\mathbb{P}(\mathbf{X}_t) = \Pr(\mathbf{X})$ for all $t > 1$. Equation 15.16, the detailed balance property, is

$$\Pr(\mathbf{x})\mathbb{P}(\mathbf{X}_i = \mathbf{x}' \mid \mathbf{X}_{i-1} = \mathbf{x}) = \Pr(\mathbf{x}')\mathbb{P}(\mathbf{X}_i = \mathbf{x} \mid \mathbf{X}_{i-1} = \mathbf{x}'),$$

required to hold for all states $\mathbf{x}, \mathbf{x}'$.

Solution. Summing Equation 15.16 over $\mathbf{x}$ gives stationarity. Write $K(\mathbf{x}, \mathbf{x}') = \mathbb{P}(\mathbf{X}_i = \mathbf{x}' \mid \mathbf{X}_{i-1} = \mathbf{x})$ for the common transition matrix (the chain is homogeneous) and set $\mathbb{P}(\mathbf{X}_1) = \Pr(\mathbf{X})$. If $\mathbb{P}(\mathbf{X}_t = \mathbf{x}) = \Pr(\mathbf{x})$ for all $\mathbf{x}$, then marginalizing $\mathbf{X}_t$ out of the joint of $\mathbf{X}_t, \mathbf{X}_{t+1}$,

$$\begin{aligned}
\mathbb{P}(\mathbf{X}_{t+1} = \mathbf{x}') &= \sum_{\mathbf{x}} \mathbb{P}(\mathbf{X}_t = \mathbf{x})\, K(\mathbf{x}, \mathbf{x}') = \sum_{\mathbf{x}} \Pr(\mathbf{x}) K(\mathbf{x}, \mathbf{x}') \\
&= \sum_{\mathbf{x}} \Pr(\mathbf{x}') K(\mathbf{x}', \mathbf{x}) \qquad \text{(Equation 15.16)} \\
&= \Pr(\mathbf{x}') \sum_{\mathbf{x}} K(\mathbf{x}', \mathbf{x}) = \Pr(\mathbf{x}'),
\end{aligned}$$

the final row sum being 1 as a normalized CPT column for $\mathbf{X}_{t+1}$. Induction from the base case $t = 1$ gives $\mathbb{P}(\mathbf{X}_t) = \Pr(\mathbf{X})$ for every t.

Problem 15.27. Provide a Markov chain over three states that has a stationary distribution yet does not satisfy the detailed balance property given in Equation 15.16.

Solution. Take the states $\{1, 2, 3\}$ with the homogeneous transition matrix $K(\mathbf{x}, \mathbf{x}') = \mathbb{P}(\mathbf{X}_i = \mathbf{x}' \mid \mathbf{X}_{i-1} = \mathbf{x})$ given by the biased rotation

from \ to	1	2	3
1	0	2/3	1/3
2	1/3	0	2/3
3	2/3	1/3	0

one step clockwise ($1 \to 2 \to 3 \to 1$) with probability $2/3$, one step counterclockwise with probability $1/3$; each row sums to 1 (Check!).
The uniform $\Pr(1) = \Pr(2) = \Pr(3) = 1/3$ is stationary, since each $\mathbf{x}'$ receives mass from exactly two states, of weights $2/3$ and $1/3$:

$$\sum_{\mathbf{x}} \Pr(\mathbf{x}) K(\mathbf{x}, \mathbf{x}') = \tfrac{1}{3} \cdot \tfrac{2}{3} + \tfrac{1}{3} \cdot \tfrac{1}{3} = \tfrac{1}{3} = \Pr(\mathbf{x}'),$$

one-step invariance propagating to all t by the induction of Exercise 15.26. Detailed balance fails already at $\mathbf{x} = 1$, $\mathbf{x}' = 2$:

$$\Pr(1)K(1,2) = \tfrac{1}{3} \cdot \tfrac{2}{3} = \tfrac{2}{9} \neq \tfrac{1}{9} = \tfrac{1}{3} \cdot \tfrac{1}{3} = \Pr(2)K(2,1).$$

Problem 15.28. The Gibbs sampler can be implemented in two ways. In one case, the variable S whose state will be sampled is chosen at random. In another case, a particular sequence for variables S is predetermined and sampling is always performed according to that sequence. The second method is known to result in an irreducible Markov chain that does not satisfy the detailed balance property. Compare the performance of these methods empirically.
(The first method is the transition matrix of Definition 15.6, the random-scan Gibbs sampler implemented by Algorithm 48, whose line 1 chooses S uniformly at random from $\mathbf{X}$. The second, systematic-scan or sweep, sampler replaces that line by "let S be the next variable in a fixed cyclic order".)

Solution. The systematic scan wins: at equal work its standard error is about 1.35 times smaller, hence its variance about 1.83 times smaller. The test network is Figure 15.1, with A (winter), B (sprinkler), C (rain), D (wet grass), E (slippery road), edges $A \to B$, $A \to C$, $B \to D$, $C \to D$, $C \to E$, and CPTs

A	θ_A
true	.6
false	.4

A	B	$\theta_{B\|A}$
true	true	.2
true	false	.8
false	true	.75
false	false	.25

A	C	$\theta_{C\|A}$
true	true	.8
true	false	.2
false	true	.1
false	false	.9

B	C	D	$\theta_{D\|BC}$
true	true	true	.95
true	true	false	.05
true	false	true	.9
true	false	false	.1
false	true	true	.8
false	true	false	.2
false	false	true	0
false	false	false	1

C	E	$\theta_{E\mid C}$
true	true	.7
true	false	.3
false	true	0
false	false	1

Take evidence $\mathbf{e} = d$ and sample $\mathbf{X} = \{A, B, C, E\}$ from $\Pr(S \mid \mathbf{x} - S, d)$ per Theorem 15.17. Exact enumeration gives $\Pr(d) = .6995$ and

$$\Pr(a \mid d) = .600429, \qquad \Pr(b \mid d) = .549392,$$
$$\Pr(c \mid d) = .621730, \qquad \Pr(e \mid d) = .435211.$$

Here $\theta_{e|\bar{c}} = 0$ and $\theta_{d|\bar{b}\bar{c}} = 0$ kill six states, but both chains are irreducible on the remaining 10, so Theorem 15.15 applies to each. Protocol: one unit of work is one single-variable resampling (the systematic sampler cycling $A, B, C, E, \ldots$); each run starts uniformly on the 10 support states, burns in 400 updates, records the next N, and is repeated $R = 2000$ times. Root-mean-square error for $\Pr(a \mid d)$ against N:

N	random scan	systematic scan	ratio
250	.2649	.2061	1.285
1000	.1457	.1111	1.312
4000	.0762	.0561	1.358

and for all four queries at $N = 4000$:

query	random scan	systematic scan	ratio
$\Pr(a \mid d)$	.0762	.0561	1.358
$\Pr(b \mid d)$	.0679	.0500	1.358
$\Pr(c \mid d)$	.0829	.0615	1.348
$\Pr(e \mid d)$	.0639	.0470	1.360

Both samplers are unbiased to within .0017 at these lengths and both errors decay like $N^{-1/2}$ per Theorem 15.15, so the whole difference is the constant: reaching a given accuracy costs the random scan roughly 83% more updates. Two facts account for it.

(i) Stationary but not reversible taking one sweep as the transition, $K^{\mathrm{sw}} = K_A K_B K_C K_E$, each K_X preserves $\Pr(\mathbf{X} \mid d)$ by Theorem 15.16 and hence so does the product, but the reversal is the opposite sweep $K_E K_C K_B K_A \neq K^{\mathrm{sw}}$. Measured, $\max|\Pr(\mathbf{x})K(\mathbf{x}, \mathbf{x}') - \Pr(\mathbf{x}')K(\mathbf{x}', \mathbf{x})|$ is 3.5×10^{-18} for the random-scan kernel but .014883 for K^{sw}; each kernel has 1 as a simple eigenvalue with $\Pr K - \Pr$ below 6×10^{-17}, confirming the shared unique stationary distribution. This is Exercise 15.27 concretely.

(ii) Larger spectral gap the second eigenvalue modulus per update is .98333 for the random scan against $.88112^{1/4} = .96886$ for the sweep, predicting

$$\frac{(1 + \lambda_{\mathrm{ran}})/(1 - \lambda_{\mathrm{ran}})}{(1 + \lambda_{\mathrm{sys}})/(1 - \lambda_{\mathrm{sys}})} = \frac{119.0}{63.2} \approx 1.88,$$

close to the measured 1.83; over 4 updates the random scan leaves a given variable untouched with probability $(3/4)^4 \approx .32$, the sweep never.

14.5 Exercises 15.29–15.31

Problem 15.29. A Markov chain is said to be *aperiodic* if and only if there exists a specific time $t > 1$ such that for every pair of states x and x' we have $P(X_t = x' \mid X_1 = x) > 0$. (The literature on Markov chains contains multiple definitions of the notion of aperiodicity; this is the one used here.) If a Markov chain is aperiodic, it will also be irreducible and hence have a unique stationary distribution, say $Pr(X)$. (Note the subtle difference between aperiodicity and irreducibility, where the latter states that for every pair of states x and x' there exists a specific time $t > 1$, possibly depending on the pair, such that $P(X_t = x' \mid X_1 = x) > 0$. As defined here, aperiodicity implies irreducibility but the converse is not true.) Moreover, if the chain is aperiodic it will converge to its stationary distribution, that is,

$$\lim_{t\to\infty} P(X_t = x' \mid X_1 = x) \;=\; Pr(x')$$

for all states x and x'. Hence, when simulating the Markov chain, the simulated instantiations are eventually sampled from the stationary distribution $Pr(X)$ and become independent of the initial state at time 1.

Consider now the Markov chain for a binary variable X with the transition matrix $P(x \mid \bar{x}) = 1$ and $P(\bar{x} \mid x) = 1$; hence $P(\bar{x} \mid \bar{x}) = 0$ and $P(x \mid x) = 0$. Is this chain aperiodic? Is it irreducible? If it is, identify its unique stationary distribution. Will the chain converge to any distribution?

Solution. Not aperiodic, but irreducible, with the uniform distribution as its unique stationary distribution, and it converges to nothing. The transition matrix, rows indexed by the state at time $i-1$ and columns by the state at time i, is

$P(X_i \mid X_{i-1})$	$X_i = x$	$X_i = \bar{x}$
$X_{i-1} = x$	0	1
$X_{i-1} = \bar{x}$	1	0

The chain flips X deterministically at every step, so by induction on t (if $X_t = y$ with probability 1 then $X_{t+1} = \bar{y}$ with probability 1),

$$P(X_t = x \mid X_1 = x) = \mathbf{1}[t \text{ odd}], \qquad P(X_t = \bar{x} \mid X_1 = x) = \mathbf{1}[t \text{ even}].$$

(i) Aperiodicity fails it demands a *single* $t > 1$ good for all pairs, yet at each fixed t exactly one of the two displayed probabilities is 0.

(ii) Irreducibility holds the time may depend on the pair, and $t = 2$ gives $P(X_2 = \bar{x} \mid X_1 = x) = 1$ while $t = 3$ gives $P(X_3 = x \mid X_1 = x) = 1$, for either choice of x.

(iii) Unique stationary distribution writing $Pr(x) = p$, one step gives $P(X_i = x) = p \cdot 0 + (1 - p) \cdot 1 = 1 - p$, so stationarity forces $p = 1/2$ and $Pr(x) = Pr(\bar{x}) = 1/2$.

(iv) Convergence fails $P(X_t = x \mid X_1 = x)$ is $1, 0, 1, 0, \ldots$, which has no limit, so X_t never becomes independent of X_1.

Problem 15.30. Show that the Gibbs transition matrix given in Definition 15.6 satisfies the following condition:

$$\sum_{x'} P(X_i = x' \mid X_{i-1} = x) \;=\; 1.$$

Recall Definition 15.6: given a distribution $Pr(X)$ over m variables X, the corresponding Gibbs transition matrix is

$$P(X_i = x' \mid X_{i-1} = x) = \begin{cases} 0, & \text{if } x \text{ and } x' \text{ disagree on more than one variable} \\ \dfrac{1}{m} Pr(s' \mid x - S), & \text{if } x \text{ and } x' \text{ disagree on a single variable } S, \\ & \text{which has value } s' \text{ in } x' \\ \dfrac{1}{m} \displaystyle\sum_{S \in X} Pr(s_x \mid x - S), & \text{if } x = x' \text{ and } s_x \text{ is the value of } S \text{ in } x. \end{cases}$$

Solution. The row splits into m full distributions $Pr(S \mid x - S)$, each summing to 1. Fix the row x, write $X = \{S_1, \ldots, S_m\}$, let s_x^S be the value of S in x, and partition the states x' by how many variables they disagree with x on. Two or more contributes 0 by the first case of Definition 15.6, and $(S, s') \mapsto (x - S, s')$ over $S \in X$, $s' \neq s_x^S$, is a bijection onto the states disagreeing on exactly one (Check!), so with the remaining two cases, writing $P(x' \mid x) = P(X_i = x' \mid X_{i-1} = x)$,

$$\sum_{x'} P(x' \mid x) = 0 + \frac{1}{m} \sum_{S \in X} \Big[\sum_{s' \neq s_x^S} Pr(s' \mid x - S) + Pr(s_x^S \mid x - S) \Big]$$
$$= \frac{1}{m} \sum_{S \in X} \sum_{s'} Pr(s' \mid x - S) = \frac{1}{m} \sum_{S \in X} 1 \;=\; 1,$$

the inner sums merging as they range over $s' \neq s_x^S$ and $s' = s_x^S$, and each $Pr(S \mid x - S)$ summing to 1 over the values of S (defined, since Definition 15.6 requires $Pr(x - S) > 0$).

Problem 15.31. Consider the Gibbs transition matrix $P(X_i \mid X_{i-1})$ given in Definition 15.6 and let x be a state of variables X. Consider now the state x' generated as follows:

- Let S be a variable chosen randomly from X.
- Let s be a state of variable S sampled from $Pr(S \mid X - S)$, that is, from $Pr(S \mid x - S)$, where the variables $X - S$ are fixed to the values they take in the state x.
- Set state x' to $x - S, s$.

Show that the state x' is generated with probability $P(X_i = x' \mid X_{i-1} = x)$.
Recall Definition 15.6: given a distribution $Pr(X)$ over m variables X, the corresponding Gibbs transition matrix is

$$P(X_i = x' \mid X_{i-1} = x) = \begin{cases} 0, & \text{if } x \text{ and } x' \text{ disagree on more than one variable} \\ \dfrac{1}{m} Pr(s' \mid x - S), & \text{if } x \text{ and } x' \text{ disagree on a single variable } S, \\ & \text{which has value } s' \text{ in } x' \\ \dfrac{1}{m} \displaystyle\sum_{S \in X} Pr(s_x \mid x - S), & \text{if } x = x' \text{ and } s_x \text{ is the value of } S \text{ in } x. \end{cases}$$

Here *chosen randomly* means chosen uniformly at random from the m variables in X.

Solution. The two-stage draw picks the pair (S, s) with probability $Q(S, s) = \frac{1}{m} Pr(s \mid x - S)$ by the chain rule and outputs the deterministic function $g(S, s) = (x - S, s)$, so by total probability over the disjoint outcomes, writing s_x^S for the value of S in x and $X = \{S_1, \ldots, S_m\}$,

$$Q(x') \;=\; \sum_{S \in X} \sum_{s} \frac{1}{m} Pr(s \mid x - S) \cdot \mathbf{1}\big[(x - S, s) = x'\big] \,.$$

Since $g(S, s)$ agrees with x off S, it disagrees with x on at most one variable, and on S exactly when $s \neq s_x^S$, splitting Q into the three cases of Definition 15.6:

(i) x' **disagrees with** x **on more than one variable** no outcome maps to x', so $Q(x') = 0$.

(ii) x' **disagrees exactly on** S_0**, with value** s' **there** any $S \neq S_0$ leaves S_0 at its value in x, so only $S = S_0$ can match, and then only for $s = s'$; hence $Q(x') = \frac{1}{m} Pr(s' \mid x - S_0)$, the conditioning instantiation being the one named in the definition because $x - S_0 = x' - S_0$.

(iii) $x' = x$ $g(S, s) = x$ iff $s = s_x^S$, one value of s for each of the m choices of S, so $Q(x) = \frac{1}{m} \sum_{S \in X} Pr(s_x^S \mid x - S)$.

The cases being exhaustive, $Q(x') = P(X_i = x' \mid X_{i-1} = x)$ throughout (assuming, as in Exercise 15.30, $Pr(x - S) > 0$ for every S).

15 Sensitivity Analysis

Contents

15.1 Exercises 16.1–16.7 270
15.2 Exercises 16.8–16.14 274
15.3 Exercises 16.15–16.17 281

15.1 Exercises 16.1–16.7

Problem 16.1. Let us say that two probability distributions $\Pr^0(\mathbf{X})$ and $\Pr(\mathbf{X})$ have the same support if for every instantiation $\mathbf{x}$, $\Pr^0(\mathbf{x}) = 0$ iff $\Pr(\mathbf{x}) = 0$. Show that the CD distance satisfies the following property: $D(\Pr^0, \Pr) = \infty$ iff the distributions $\Pr^0$ and $\Pr$ do not have the same support.
Recall Definition 16.1: for two distributions over the same set of variables $\mathbf{X}$,

$$D(\Pr^0, \Pr) \;=\; \ln \max_{\mathbf{x}} \frac{\Pr(\mathbf{x})}{\Pr^0(\mathbf{x})} \;-\; \ln \min_{\mathbf{x}} \frac{\Pr(\mathbf{x})}{\Pr^0(\mathbf{x})},$$

where $0/0 \stackrel{\text{def}}{=} 1$ and $\infty/\infty \stackrel{\text{def}}{=} 1$.

Solution. $D = \infty$ exactly when the ratio $r(\mathbf{x}) = \Pr(\mathbf{x})/\Pr^0(\mathbf{x})$ of Definition 16.1 hits ∞ or 0, and under the convention $0/0 = 1$ those are precisely the support disagreements:

$\Pr^0(\mathbf{x})$	$\Pr(\mathbf{x})$	$r(\mathbf{x})$
$= 0$	$= 0$	1
> 0	> 0	finite and positive
$= 0$	> 0	∞
> 0	$= 0$	0

No $\infty - \infty$ can arise, since $\min_{\mathbf{x}} r = \infty$ would force $\Pr^0 \equiv 0$ and $\max_{\mathbf{x}} r = 0$ would force $\Pr \equiv 0$, each against normalization; so $\max r \in (0, \infty]$, $\min r \in [0, \infty)$ and $D \in (-\infty, +\infty]$.

(i) Different support some $\mathbf{x}$ falls in row three, giving $\max r = \infty$, or in row four, giving $\min r = 0$; either way $D(\Pr^0, \Pr) = \infty$.

(ii) Same support rows three and four are empty, so $r = 1$ off the common support and $r \in (0, \infty)$ on it; over finitely many instantiations $0 < \min_{\mathbf{x}} r \le \max_{\mathbf{x}} r < \infty$, whence $D(\Pr^0, \Pr) < \infty$.

Problem 16.2. Show that the CD distance satisfies the properties of positiveness, symmetry, and triangle inequality. That is, if $\Pr^0$, $\Pr$, and $\Pr'$ are three probability distributions over the same set of variables $\mathbf{X}$, then
Positiveness $D(\Pr^0, \Pr) \ge 0$, and $D(\Pr^0, \Pr) = 0$ iff $\Pr^0 = \Pr$.
Symmetry $D(\Pr^0, \Pr) = D(\Pr, \Pr^0)$.
Triangle inequality $D(\Pr^0, \Pr) + D(\Pr, \Pr') \ge D(\Pr^0, \Pr')$.
Here D is the CD distance of Definition 16.1,

$$D(\Pr^0, \Pr) \;=\; \ln \max_{\mathbf{x}} \frac{\Pr(\mathbf{x})}{\Pr^0(\mathbf{x})} \;-\; \ln \min_{\mathbf{x}} \frac{\Pr(\mathbf{x})}{\Pr^0(\mathbf{x})},$$

with $0/0 = 1$ and $\infty/\infty = 1$.

Solution. All three rest on the sandwich $\min_{\mathbf{x}} r \le 1 \le \max_{\mathbf{x}} r$, where $r(\mathbf{x}) = \Pr(\mathbf{x})/\Pr^0(\mathbf{x})$ as in Exercise 16.1.

(i) Positiveness if $r(\mathbf{x}) > 1$ for every $\mathbf{x}$, the case table of Exercise 16.1 excludes its first and fourth rows, so $\Pr(\mathbf{x}) > \Pr^0(\mathbf{x})$ throughout and summing gives $1 > 1$; symmetrically $r(\mathbf{x}) < 1$ throughout excludes the first and third rows and gives $1 < 1$. This is the sandwich, whence

$$D(\Pr^0, \Pr) = \ln \max_{\mathbf{x}} r - \ln \min_{\mathbf{x}} r \;\ge\; \ln 1 - \ln 1 \;=\; 0.$$

And $D = 0$ forces $\max r = \min r$, hence $r \equiv 1$ by the sandwich, hence $\Pr = \Pr^0$ by rows one and two of the table; conversely $\Pr = \Pr^0$ gives $r \equiv 1$ and $D = 0$.

(ii) Symmetry the conventions are reciprocal-consistent, so the ratio $\tilde{r} = \Pr^0/\Pr$ used for $D(\Pr, \Pr^0)$ equals $1/r$ in the extended reals on each of the four rows (Check!), and $t \mapsto 1/t$ is order-reversing on $[0, \infty]$, so $\max \tilde{r} = 1/\min r$ and $\min \tilde{r} = 1/\max r$ and

$$D(\Pr, \Pr^0) = -\ln \min_{\mathbf{x}} r + \ln \max_{\mathbf{x}} r = D(\Pr^0, \Pr),$$

the rearrangement legitimate because at most one of the two logarithms is infinite.

(iii) Triangle inequality trivial if either left distance is ∞, both being nonnegative by (i), so assume both finite; then all three distributions share one support S by Exercise 16.1. Put $a = \Pr/\Pr^0$, $b = \Pr'/\Pr$, $c = \Pr'/\Pr^0$. Cancelling $\Pr(\mathbf{x})$ gives $c = ab$ on S, where all six probabilities are positive, and off S the convention $0/0 = 1$ makes it read $1 = 1 \cdot 1$. So $c(\mathbf{x}) = a(\mathbf{x})b(\mathbf{x}) \in (0, \infty)$ throughout, no indeterminate product, giving $\max c \le (\max a)(\max b)$ and $\min c \ge (\min a)(\min b)$ between positive reals, so logarithms preserve these and

$$\begin{aligned} D(\Pr^0, \Pr') &= \ln \max_{\mathbf{x}} c - \ln \min_{\mathbf{x}} c \\ &\le \big(\ln \max a - \ln \min a\big) + \big(\ln \max b - \ln \min b\big) \\ &= D(\Pr^0, \Pr) + D(\Pr, \Pr'). \end{aligned}$$

Problem 16.3. Consider a distribution $\Pr(\mathbf{X})$, event α, and show that

$$\frac{\Pr(\alpha)}{\Pr^0(\alpha)} \;\le\; \max_{\mathbf{x} \models \alpha} \frac{\Pr(\mathbf{x})}{\Pr^0(\mathbf{x})}$$

and

$$\frac{\Pr(\alpha)}{\Pr^0(\alpha)} \;\ge\; \min_{\mathbf{x} \models \alpha} \frac{\Pr(\mathbf{x})}{\Pr^0(\mathbf{x})}.$$

Here $\Pr^0(\mathbf{X})$ is a second distribution over the same variables, $\mathbf{x}$ ranges over the instantiations of $\mathbf{X}$ that satisfy α, and the conventions of Definition 16.1 apply ($0/0 = 1$).

Solution. Both bounds are the mediant inequality, that a ratio of sums lies between the extreme ratios of its terms. Let α be satisfiable, $A = \{\mathbf{x} : \mathbf{x} \models \alpha\}$, and, with $r = \Pr/\Pr^0 \in [0, \infty]$ by the case table of Exercise 16.1,

$$M = \max_{\mathbf{x} \in A} r(\mathbf{x}), \qquad m = \min_{\mathbf{x} \in A} r(\mathbf{x}).$$

Since A need not carry all the mass, the normalization argument of Exercise 16.1 does not exclude $M = 0$ or $m = \infty$, so those are handled explicitly.

(i) Upper bound trivial if $M = \infty$, so let $M < \infty$. Then $\Pr(\mathbf{x}) \le M \Pr^0(\mathbf{x})$ on A: where $\Pr^0(\mathbf{x}) > 0$ this is the definition of M, and where $\Pr^0(\mathbf{x}) = 0$ finiteness of M forces $\Pr(\mathbf{x}) = 0$, leaving $0 \le 0$. Summing over A gives $\Pr(\alpha) \le M \Pr^0(\alpha)$, and dividing by $\Pr^0(\alpha) > 0$ gives the claim. If instead $\Pr^0(\alpha) = 0$, then $\Pr(\alpha) = 0$ too, so the left-hand side is $0/0 = 1$ while every ratio on A is $0/0 = 1$ and $M = 1$: equality.

(ii) Lower bound trivial if $m = 0$. If $m = \infty$ then $\Pr^0(\mathbf{x}) = 0 < \Pr(\mathbf{x})$ throughout A, so $\Pr^0(\alpha) = 0 < \Pr(\alpha)$ and the claim reads $\infty \ge \infty$. For $0 < m < \infty$ we get $\Pr(\mathbf{x}) \ge m \Pr^0(\mathbf{x})$ on A, the case $\Pr^0(\mathbf{x}) = 0$ reading $\Pr(\mathbf{x}) \ge m \cdot 0 = 0$, a legitimate product since m is finite; summing and dividing by $\Pr^0(\alpha) > 0$ gives the claim. If $\Pr^0(\alpha) = 0$ the left-hand side is ∞ when $\Pr(\alpha) > 0$, and otherwise every ratio on A is 1, so $m = 1$ and the claim again holds.

Problem 16.4. Show that the bounds given by Theorem 16.1 are tight in the sense that for every pair of distributions $\Pr^0$ and $\Pr$, there are events α and β such that

$$\frac{O(\alpha \mid \beta)}{O^0(\alpha \mid \beta)} \;=\; e^{D(\Pr^0, \Pr)}$$

$$\frac{O(\neg\alpha \mid \beta)}{O^0(\neg\alpha \mid \beta)} \;=\; e^{-D(\Pr^0, \Pr)}.$$

Recall that Theorem 16.1 states $e^{-D(\Pr^0,\Pr)} \le O(\alpha|\beta)/O^0(\alpha|\beta) \le e^{D(\Pr^0,\Pr)}$ for arbitrary events α and β, where $O(\alpha|\beta) = \Pr(\alpha|\beta)/\Pr(\neg\alpha|\beta)$.

Solution. Take $\alpha = \mathbf{x}^\star$ and $\beta = \mathbf{x}^\star \vee \mathbf{x}_\star$, where $r(\mathbf{x}^\star) = \max_{\mathbf{x}} r$ and $r(\mathbf{x}_\star) = \min_{\mathbf{x}} r$ for $r = \Pr/\Pr^0$. Distinct instantiations being mutually exclusive, $\alpha \wedge \beta = \mathbf{x}^\star$ and $\neg\alpha \wedge \beta = \mathbf{x}_\star$, and the odds conditioned on β are ratios of joint probabilities.

(i) $0 < D < \infty$ by Exercise 16.1 the supports coincide, and $r = 1$ off the common support S. If $\max r = 1$ then $\Pr \le \Pr^0$ on S, strictly somewhere since $\min r < 1$ by Exercise 16.2 and $D > 0$, so summing over S gives $1 < 1$; hence $\max r > 1$ and it is attained inside S, and symmetrically $\min r < 1$ inside S. So all four of $\Pr(\mathbf{x}^\star), \Pr^0(\mathbf{x}^\star), \Pr(\mathbf{x}_\star), \Pr^0(\mathbf{x}_\star)$ are positive and $\mathbf{x}^\star \neq \mathbf{x}_\star$, so every odds below is finite and positive, and

$$\frac{O(\alpha|\beta)}{O^0(\alpha|\beta)} = \frac{\Pr(\mathbf{x}^\star)}{\Pr(\mathbf{x}_\star)} \cdot \frac{\Pr^0(\mathbf{x}_\star)}{\Pr^0(\mathbf{x}^\star)} = \frac{r(\mathbf{x}^\star)}{r(\mathbf{x}_\star)} = e^{D(\Pr^0,\Pr)},$$

whence $O(\neg\alpha|\beta)/O^0(\neg\alpha|\beta)$, the reciprocal, is e^{-D}.

(ii) $D = 0$ then $\Pr = \Pr^0$ by Exercise 16.2 and both identities read $1 = e^0$; it remains to name a pair whose odds ratio is unambiguously 1. If two instantiations $\mathbf{x}_1 \neq \mathbf{x}_2$ have positive probability, $\alpha = \mathbf{x}_1$, $\beta = \mathbf{x}_1 \vee \mathbf{x}_2$ makes both odds the positive real $\Pr(\mathbf{x}_1)/\Pr(\mathbf{x}_2)$. Otherwise Pr is a point mass at $\mathbf{x}_1$, and for a second instantiation $\mathbf{x}_2$ take $\alpha = \mathbf{x}_2$, $\beta = \mathbf{x}_1 \vee \mathbf{x}_2$: both odds are 0 with ratio $0/0 = 1$, and both $\neg\alpha$ odds are ∞ with ratio $\infty/\infty = 1$. With only one instantiation the claim is vacuous.

(iii) $D = \infty$ the supports differ by Exercise 16.1. If $\Pr^0(\mathbf{x}^\star) = 0 < \Pr(\mathbf{x}^\star)$ for some $\mathbf{x}^\star$, pick $\mathbf{x}_\star \neq \mathbf{x}^\star$ with $\Pr^0(\mathbf{x}_\star) > 0$ (one exists as $\Pr^0$ sums to one); then $O^0(\alpha|\beta) = 0$ while $O(\alpha|\beta) > 0$, so the ratio is $\infty = e^D$ and its reciprocal $0 = e^{-D}$. If instead $\Pr(\mathbf{x}_\star) = 0 < \Pr^0(\mathbf{x}_\star)$, pick $\mathbf{x}^\star \neq \mathbf{x}_\star$ with $\Pr(\mathbf{x}^\star) > 0$; then $O(\alpha|\beta) = \infty$ while $O^0(\alpha|\beta)$ is finite, and the same conclusion follows.

Problem 16.5. Show that Inequality 16.2 is implied by Theorem 16.1. That is, derive

$$\frac{e^{-d}\,p}{\big(e^{-d}-1\big)p+1} \;\le\; \Pr(\alpha|\beta) \;\le\; \frac{e^{d}\,p}{\big(e^{d}-1\big)p+1},$$

where $p = \Pr^0(\alpha|\beta)$ and $d = D(\Pr^0, \Pr)$, from the odds bound of Theorem 16.1,

$$e^{-d} \;\le\; \frac{O(\alpha|\beta)}{O^0(\alpha|\beta)} \;\le\; e^{d},$$

in which $O(\alpha|\beta) = \Pr(\alpha|\beta)/\Pr(\neg\alpha|\beta)$ and $O^0(\alpha|\beta) = \Pr^0(\alpha|\beta)/\Pr^0(\neg\alpha|\beta)$.

Solution. Apply the strictly increasing map $f(t) = t/(1+t)$ on $[0,\infty]$, inverse to the odds map $q \mapsto q/(1-q)$ with $f(\infty) = 1$ (Check!), to each half of Theorem 16.1. Write $p = \Pr^0(\alpha|\beta)$, $q = \Pr(\alpha|\beta)$, so that $O^0(\alpha|\beta) = p/(1-p)$ and $O(\alpha|\beta) = q/(1-q)$, and assume $\Pr^0(\beta), \Pr(\beta) > 0$ and $d < \infty$ (at $d = \infty$ Theorem 16.1 constrains the odds not at all, and Inequality 16.2, whose $e^{\pm d}p$ is then indeterminate at $p = 0$, is read as the vacuous $0 \le \Pr(\alpha|\beta) \le 1$).

(i) Upper bound Theorem 16.1 gives $q/(1-q) \le e^d p/(1-p) =: K$. If $K = \infty$, that is $p = 1$, the claimed bound is $e^d/e^d = 1 \ge q$. Otherwise K is a nonnegative real, multiplying by $1-q \ge 0$ gives $q(1+K) \le K$, so $q \le f(K)$ (here $q = 1$ is impossible, as it would make the odds infinite against finite K), and

$$f(K) = \frac{e^d p/(1-p)}{1 + e^d p/(1-p)} = \frac{e^d p}{(1-p) + e^d p} = \frac{e^d p}{\big(e^d-1\big)p+1}.$$

(ii) Lower bound symmetrically $q/(1-q) \ge e^{-d}p/(1-p) =: k$. If $k = \infty$ then $p = q = 1$ and the claimed bound is $1 = q$; otherwise applying f to both sides gives

$$q \;\ge\; f(k) = \frac{e^{-d}p}{(1-p) + e^{-d}p} = \frac{e^{-d}p}{\big(e^{-d}-1\big)p+1}.$$

Both denominators $(1-p) + e^{\pm d}p$ are positive on $[0,1]$ (p and $1-p$ are not both zero), so the two displays are Inequality 16.2.

Problem 16.6. Consider two Bayesian networks $\mathcal{N}^0$ and $\mathcal{N}$, where network $\mathcal{N}$ is obtained from $\mathcal{N}^0$ by changing a single parameter value θ^0_x in the CPT of root node X (co-varying parameters being changed according to the proportional scheme of Definition 16.2). Show the following for any evidence $\mathbf{e}$,

$$\frac{O(x|\mathbf{e})}{O^0(x|\mathbf{e})} \;=\; \frac{O(x)}{O^0(x)}.$$

Here $O(x|\mathbf{e}) = \Pr(x|\mathbf{e})/\Pr(\neg x|\mathbf{e})$ and $O(x) = \Pr(x)/\Pr(\neg x)$ are the posterior and prior odds of x in $\mathcal{N}$, and similarly for $\mathcal{N}^0$.

Solution. The evidence cancels because the proportional scheme scales every co-varying parameter by one common factor $c = (1-\theta_x)/(1-\theta^0_x)$. Let $\mathbf{Y}$ be the other network variables, with $\mathbf{e}$ an instantiation of variables among them (if $\mathbf{e}$ fixed X, both posterior odds would degenerate simultaneously). Only Θ_X changes, so writing $f(x^\star, \mathbf{y})$ for the product of the CPT entries of all variables other than X and $g(x^\star) = \sum_{\mathbf{y}\models\mathbf{e}} f(x^\star, \mathbf{y})$, which is therefore the same in both networks,

$$\Pr(x^\star, \mathbf{e}) = \theta_{x^\star}\, g(x^\star), \qquad \Pr^0(x^\star, \mathbf{e}) = \theta^0_{x^\star}\, g(x^\star).$$

Definition 16.2, with $\theta^0_x \neq 1$ (else $\Pr^0(\neg x, \mathbf{e}) = 0$ and the initial odds are infinite), gives $\theta_{x^\star} = c\,\theta^0_{x^\star}$ for every $x^\star \neq x$, hence

$$\Pr(\neg x, \mathbf{e}) = \sum_{x^\star \neq x} \theta_{x^\star}\, g(x^\star) = c \cdot \Pr^0(\neg x, \mathbf{e}).$$

Posterior odds are ratios of joint probabilities, $\Pr(\mathbf{e})$ cancelling, so assuming $g(x) > 0$, $\Pr^0(\neg x, \mathbf{e}) > 0$ and $\theta_x < 1$ (if $\theta_x = 1$ then $c = 0$ and both $O(x|\mathbf{e})$ and $O(x)$ are infinite, so the identity reads $\infty = \infty$),

$$\begin{aligned}\frac{O(x|\mathbf{e})}{O^0(x|\mathbf{e})} &= \frac{\theta_x\, g(x)}{c\,\Pr^0(\neg x, \mathbf{e})} \cdot \frac{\Pr^0(\neg x, \mathbf{e})}{\theta^0_x\, g(x)} = \frac{1}{c} \cdot \frac{\theta_x}{\theta^0_x} \\ &= \frac{\theta_x/(1-\theta_x)}{\theta^0_x/(1-\theta^0_x)} = \frac{O(x)}{O^0(x)},\end{aligned}$$

the last step because X is a root, so its prior is its CPT: $\Pr(x) = \theta_x$ and $\Pr(\neg x) = 1 - \theta_x$.

Problem 16.7. Show that $D(\Theta^0_{X|\mathbf{U}}, \Theta_{X|\mathbf{U}}) \geq \max_{\mathbf{u}} D(\Theta^0_{X|\mathbf{u}}, \Theta_{X|\mathbf{u}})$.
Here $\Theta^0_{X|\mathbf{U}}$ and $\Theta_{X|\mathbf{U}}$ are an initial and a modified CPT for variable X with parents $\mathbf{U}$, $\Theta^0_{X|\mathbf{u}}$ and $\Theta_{X|\mathbf{u}}$ are the rows of these CPTs corresponding to the parent instantiation $\mathbf{u}$, and the CD distance of Definition 16.1 is applied to tables by taking the extreme entry-wise ratios over the entries of the tables in question:

$$D(\Theta^0_{X|\mathbf{U}}, \Theta_{X|\mathbf{U}}) = \ln \max_{x,\mathbf{u}} \frac{\theta_{x|\mathbf{u}}}{\theta^0_{x|\mathbf{u}}} - \ln \min_{x,\mathbf{u}} \frac{\theta_{x|\mathbf{u}}}{\theta^0_{x|\mathbf{u}}},$$

$$D(\Theta^0_{X|\mathbf{u}}, \Theta_{X|\mathbf{u}}) = \ln \max_{x} \frac{\theta_{x|\mathbf{u}}}{\theta^0_{x|\mathbf{u}}} - \ln \min_{x} \frac{\theta_{x|\mathbf{u}}}{\theta^0_{x|\mathbf{u}}},$$

with the conventions $0/0 = 1$ and $\infty/\infty = 1$.

Solution. The entries of one row form a subset of the CPT's entries, so its extrema are dominated. Write $r(x, \mathbf{u}) = \theta_{x|\mathbf{u}}/\theta^0_{x|\mathbf{u}} \in [0, \infty]$ and fix $\mathbf{u}$. Maximizing over a superset can only increase, minimizing over a superset can only decrease, and ln is increasing on $[0, \infty]$, so

$$\ln \max_{x,\mathbf{u}^\star} r \;\geq\; \ln \max_{x} r(x, \mathbf{u}), \qquad \ln \min_{x,\mathbf{u}^\star} r \;\leq\; \ln \min_{x} r(x, \mathbf{u}).$$

No $\infty - \infty$ arises on subtracting: every row of both CPTs sums to one, so Exercise 16.1's argument applies inside a fixed row and gives $\min_x r(x, \mathbf{u}) \leq 1 \leq \max_x r(x, \mathbf{u})$, a sandwich only widened by

extrema over the whole CPT, so every $\ln\max \geq 0$ and every $\ln\min \leq 0$. Subtracting,

$$\begin{aligned} D(\Theta^0_{X|\mathbf{U}}, \Theta_{X|\mathbf{U}}) &= \ln \max_{x,\mathbf{u}^\star} r - \ln \min_{x,\mathbf{u}^\star} r \\ &\geq \ln \max_{x} r(x,\mathbf{u}) - \ln \min_{x} r(x,\mathbf{u}) = D(\Theta^0_{X|\mathbf{u}}, \Theta_{X|\mathbf{u}}), \end{aligned}$$

and $\mathbf{u}$ was arbitrary, so the left-hand side dominates the maximum over $\mathbf{u}$. Equality would need one row realizing both global extrema, which in general fails; for binary X with binary parent U,

U	X	\(\theta^0_{x|u}\)	\(\theta_{x|u}\)	ratio
u	x	.10	.50	5.0
u	$\bar{x}$	.90	.50	$5/9 \approx 0.556$
$\bar{u}$	x	.50	.10	0.2
$\bar{u}$	$\bar{x}$	.50	.90	1.8

has row distances $\ln 5 - \ln \frac{5}{9} = \ln 9$ and $\ln 1.8 - \ln 0.2 = \ln 9$, whereas the maximum ratio 5 sits in row u and the minimum 0.2 in row $\bar{u}$:

$$D(\Theta^0_{X|U}, \Theta_{X|U}) = \ln 5 - \ln 0.2 = \ln 25 \;>\; \ln 9.$$

15.2 Exercises 16.8–16.14

Problem 16.8. Prove Inequalities 16.14, 16.15, and 16.16.
Recall the setting. Let $\mathcal{N}^0$ and $\mathcal{N}$ be two Bayesian networks over the same structure, where $\mathcal{N}$ is obtained from $\mathcal{N}^0$ by changing a single parameter value $\theta^0_{x|u}$ to a new value $\theta_{x|u}$ (co-varying parameters being adjusted by the proportional scheme of Definition 16.2), and let $\Pr^0$ and $\Pr$ be the distributions they induce. Write $\delta_{x|u} = \theta_{x|u} - \theta^0_{x|u}$. Here e is an instantiation of variables $\mathbf{E}$, and y and z are values of variables $Y, Z \notin \mathbf{E}$. By Theorem 16.4 the probability of any event α is a linear function of the new parameter value,

$$\Pr(\alpha) \;=\; \mu^\alpha_{x|u} \cdot \theta_{x|u} + \nu^\alpha_{x|u},$$

with $\mu^\alpha_{x|u}$ and $\nu^\alpha_{x|u}$ given by (16.5) and (16.6); equivalently $\Pr(\alpha) = \Pr^0(\alpha) + \mu^\alpha_{x|u} \cdot \delta_{x|u}$. The three inequalities to be proven are:

(16.14) To satisfy $\Pr(y|e) \leq \kappa$ we need a parameter change $\delta_{x|u}$ such that

$$\Pr^0(y,e) - \kappa \cdot \Pr^0(e) \;\leq\; \delta_{x|u}\left(-\mu^{y,e}_{x|u} + \kappa \cdot \mu^e_{x|u}\right).$$

(16.15) To satisfy $\Pr(y|e) - \Pr(z|e) \geq \kappa$ we need a change $\delta_{x|u}$ such that

$$\begin{aligned} &\Pr^0(y,e) - \Pr^0(z,e) - \kappa \cdot \Pr^0(e) \\ &\qquad \geq\; \delta_{x|u}\left(-\mu^{y,e}_{x|u} + \mu^{z,e}_{x|u} + \kappa \cdot \mu^e_{x|u}\right). \end{aligned}$$

(16.16) To satisfy $\Pr(y|e)/\Pr(z|e) \geq \kappa$ we need a change $\delta_{x|u}$ such that

$$\Pr^0(y,e) - \kappa \cdot \Pr^0(z,e) \;\geq\; \delta_{x|u}\left(-\mu^{y,e}_{x|u} + \kappa \cdot \mu^{z,e}_{x|u}\right).$$

Solution. Clear the denominator, substitute Theorem 16.4, and collect the $\delta_{x|u}$ terms. Assume $\Pr(e) > 0$. Linearity of $\Pr(\alpha) = \mu^\alpha_{x|u}\theta_{x|u} + \nu^\alpha_{x|u}$ in $\theta_{x|u}$ kills $\nu^\alpha_{x|u}$ on evaluating at $\theta^0_{x|u}$ and subtracting, leaving for each $\alpha \in \{y \wedge e, z \wedge e, e\}$,

$$\Pr(\alpha) = \Pr^0(\alpha) + \mu^\alpha_{x|u} \cdot \delta_{x|u}.$$

(16.14) multiplying by $\Pr(e) > 0$ preserves direction, so $\Pr(y|e) \leq \kappa$ iff $\Pr(y,e) \leq \kappa \Pr(e)$, that is

$$\Pr^0(y,e) + \mu^{y,e}_{x|u}\delta_{x|u} \;\leq\; \kappa\left(\Pr^0(e) + \mu^e_{x|u}\delta_{x|u}\right);$$

moving $\kappa \Pr^0(e)$ left and $\mu^{y,e}_{x|u}\delta_{x|u}$ right, then factoring $\delta_{x|u}$, gives 16.14 (every step an equivalence, so the condition is necessary too).

(16.15) likewise $\Pr(y|e) - \Pr(z|e) \geq \kappa$ iff $\Pr(y,e) - \Pr(z,e) \geq \kappa \Pr(e)$; substituting all three expansions and separating the $\Pr^0$ terms from the $\delta_{x|u}$ terms gives 16.15.

(16.16) here $\Pr(e)$ cancels on its own, so with $\Pr(z,e) > 0$ the constraint $\Pr(y|e)/\Pr(z|e) \geq \kappa$ is $\Pr(y,e) \geq \kappa \Pr(z,e)$; substituting the two expansions and rearranging gives 16.16.

Problem 16.9. Consider the solution space characterized by Inequality 16.17. Call a solution optimal if it minimizes the CD distance between the original and new networks. Show that optimal solutions must satisfy the equality constraint corresponding to Inequality 16.17.
For reference, Inequality 16.17 is the multiple-parameter analogue of Corollary 9. Let $\mathcal{N}^0$ and $\mathcal{N}$ be two Bayesian networks, where $\mathcal{N}$ is obtained from $\mathcal{N}^0$ by changing one parameter value $\theta^0_{x|u}$ for each parent instantiation u in the CPT of variable X (co-varying parameters being adjusted by the proportional scheme of Definition 16.2), and let $\Pr^0$ and $\Pr$ be the distributions they induce. To ensure that $\Pr(y|e) \geq \kappa$ holds, the amounts of parameter change $\delta_{x|u} = \theta_{x|u} - \theta^0_{x|u}$ must satisfy

$$\Pr^0(y,e) - \kappa \cdot \Pr^0(e) \;\geq\; \sum_u \delta_{x|u} \left(-\mu^{y,e}_{x|u} + \kappa \cdot \mu^e_{x|u} \right),$$

where $\mu^{y,e}_{x|u}$ and $\mu^e_{x|u}$ are given by (16.5).

Solution. An optimal solution must be tight: scaling a slack solution toward the origin keeps it admissible while strictly decreasing its CD distance. Abbreviate $K = \Pr^0(y,e) - \kappa \Pr^0(e)$, $c_u = -\mu^{y,e}_{x|u} + \kappa\, \mu^e_{x|u}$ and $g(\delta) = \sum_u c_u\, \delta_{x|u}$, so that Inequality 16.17 reads $g(\delta) \leq K$ and, by Corollary 10, cuts out exactly the changes enforcing $\Pr(y|e) \geq \kappa$. Query control means $K = \Pr^0(e)(\Pr^0(y|e) - \kappa) < 0$, else $\delta = 0$ already solves it at distance 0.
Only the CPT of X differs, so the chain rule cancels every other family and $\Pr(z)/\Pr^0(z) = \theta_{x_z|u_z}/\theta^0_{x_z|u_z}$, giving as in Theorem 16.2

$$D(\Pr^0, \Pr) \;=\; \ln \max_{x',u} \frac{\theta_{x'|u}}{\theta^0_{x'|u}} \;-\; \ln \min_{x',u} \frac{\theta_{x'|u}}{\theta^0_{x'|u}},$$

the extrema over parameters with $\Pr^0(x',u) > 0$; a δ lifting a zero parameter at some u with $\Pr^0(u) > 0$ has $D = \infty$, so the candidate optimum has finite distance.
For $t \in [0,1]$ put $\delta(t) = t\delta$. Then $\theta^0_{x|u} + t\delta_{x|u}$ is the convex combination $(1-t)\theta^0_{x|u} + t\theta_{x|u}$, and by Definition 16.2 so is each co-varying $\theta^0_{x^\star|u}(1 - \theta^0_{x|u} - t\delta_{x|u})/(1 - \theta^0_{x|u})$ (in the exceptional case $\theta^0_{x|u} = 1$, Definition 16.2 gives $-t\delta_{x|u}/(|X|-1) \geq 0$ since $\delta_{x|u} \leq 0$ there), so $\delta(t)$ is admissible. The ratios

$$\frac{\theta_{x|u}(t)}{\theta^0_{x|u}} = 1 + t\,\frac{\delta_{x|u}}{\theta^0_{x|u}},$$
$$\frac{\theta_{x^\star|u}(t)}{\theta^0_{x^\star|u}} = 1 - t\,\frac{\delta_{x|u}}{1 - \theta^0_{x|u}}$$

are affine in t through $(0,1)$, so for $t \geq 0$ the extremes are carried by the extreme slopes and $D(t) = \ln(1 + t\, s_{\max}) - \ln(1 + t\, s_{\min})$ with $s_{\max} \geq 0 \geq s_{\min}$, hence nondecreasing. It is strictly increasing when $\delta \neq 0$ at finite $D(\delta)$: at a u with $\delta_{x|u} \neq 0$ and $\Pr^0(u) > 0$, finiteness forces $0 < \theta^0_{x|u} < 1$ (at 1 all co-varying parameters start at 0 and any decrease of $\theta_{x|u}$ lifts one), while some co-varying $\theta^0_{x^\star|u} > 0$ as they sum to $1 - \theta^0_{x|u} > 0$, so the two displayed slopes are nonzero of opposite sign and $s_{\max} > 0 > s_{\min}$. Now let $\delta^\star$ satisfy $g(\delta^\star) < K < 0$, so $\delta^\star \neq 0$, and put $t^\star = K/g(\delta^\star) \in (0,1)$ (both terms negative with $|K| < |g(\delta^\star)|$). Linearity gives $g(t^\star\delta^\star) = K$, so $t^\star\delta^\star$ is admissible, realizes $\Pr(y|e) = \kappa$, and has strictly smaller distance. Hence every optimal δ satisfies $g(\delta) = K$.

Problem 16.10. Suppose that Pr is a distribution obtained from $\Pr^0$ by incorporating soft evidence on events $(\beta_1, \ldots, \beta_n)$ using Jeffrey's rule (see Chapter 3). Show that

$$D(\Pr^0, \Pr) \;=\; \ln \max_{i=1}^{n} \frac{\Pr(\beta_i)}{\Pr^0(\beta_i)} \;-\; \ln \min_{i=1}^{n} \frac{\Pr(\beta_i)}{\Pr^0(\beta_i)}.$$

Here $\beta_1, \ldots, \beta_n$ are mutually exclusive and exhaustive events and Jeffrey's rule (3.22) defines the new state of belief by $\Pr(\alpha) = \sum_{i=1}^{n} q_i \Pr^0(\alpha|\beta_i)$, where $q_1, \ldots, q_n$ are the new beliefs in the events $\beta_1, \ldots, \beta_n$.

Solution. The likelihood ratio is constant on each β_i, with value $k_i = \Pr(\beta_i)/\Pr^0(\beta_i)$, and these are the only ratios Definition 16.1 sees. Assume $\Pr^0(\beta_i) > 0$, as (3.22) requires.
First $q_i = \Pr(\beta_i)$: apply (3.22) with $\alpha = \beta_i$ and use mutual exclusiveness, $\Pr^0(\beta_i|\beta_j) = 0$ for $j \neq i$ and $\Pr^0(\beta_i|\beta_i) = 1$. Next, each world x satisfies exactly one β_i, so in (3.22) with $\alpha = x$ only the $j = i$ term survives and

$$\Pr(x) \;=\; q_i \frac{\Pr^0(x)}{\Pr^0(\beta_i)} \;=\; \Pr^0(x) \cdot k_i.$$

So the ratios of Definition 16.1 are exactly the k_i, each realized since $\Pr^0(\beta_i) > 0$ puts a world of positive prior probability inside β_i, plus the value $0/0 = 1$ from worlds with $\Pr^0(x) = 0$ (hence $\Pr(x) = 0$). That stray 1 moves neither extremum, the partition giving

$$\sum_{i=1}^{n} \Pr^0(\beta_i)\, k_i \;=\; \sum_{i=1}^{n} \Pr(\beta_i) \;=\; 1,$$

exhibiting 1 as a convex combination of the k_i, so $\min_i k_i \leq 1 \leq \max_i k_i$. Hence $\max_x \Pr(x)/\Pr^0(x) = \max_i k_i$ and $\min_x \Pr(x)/\Pr^0(x) = \min_i k_i$, which is the claimed formula.

Problem 16.11. Consider a simplified version of the pregnancy network in Figure 16.1 where we can only administer the scanning test to detect pregnancy. That is, the network has two nodes, P (Pregnant?, with values yes and no) and S (Scanning Test, with values $+ve$ and $-ve$), and the single edge $P \to S$. Its CPTs, taken from Figure 16.1, are

P	θ_P
yes	.87
no	.13

P	S	$\theta^0_{S\|P}$
yes	$+ve$	.90
yes	$-ve$	.10
no	$+ve$	.01
no	$-ve$	.99

so the false-negative rate of the scanning test is $\theta^0_{-ve|yes} = .10$ and its false-positive rate is $\theta^0_{+ve|no} = .01$. The current probability of pregnancy given a positive scanning test is $\Pr^0(yes\,|\,+ve) = .9983$ and the current probability of no pregnancy given a negative scanning test is $\Pr^0(no\,|\,-ve) = .5967$. We wish to apply changes in both the false-positive and false-negative rates of the scanning test by amounts of δ_1 and δ_2, respectively, such that the corresponding probabilities are at least .999 and .7 (instead of .9983 and .5967, respectively). Plot the solution spaces of δ_1 and δ_2 such that they satisfy the two constraints, and find the unique solution such that the minimum confidence levels are exactly realized by the parameter changes.

Solution. The unique exact solution is $\delta_1 = -1129/302900 = -.0037273$ and $\delta_2 = -2457/67570 = -.0363623$, the vertex where the two constraint lines meet. Both changed parameters, $\theta_{+ve|no} = .01+\delta_1$ and $\theta_{-ve|yes} = .10+\delta_2$, sit in one CPT at different parent instantiations, so Theorem 16.5 and Corollary

10 apply, and S being binary forces the co-varying $\theta_{-ve|no} = .99 - \delta_1$, $\theta_{+ve|yes} = .90 - \delta_2$:

$$\Pr(yes, +ve) = .783 - .87\,\delta_2, \qquad \Pr(no, +ve) = .0013 + .13\,\delta_1,$$
$$\Pr(no, -ve) = .1287 - .13\,\delta_1, \qquad \Pr(yes, -ve) = .087 + .87\,\delta_2,$$

which at $\delta_1 = \delta_2 = 0$ return the quoted $\Pr^0(yes|+ve) = .99834$ and $\Pr^0(no|-ve) = .59666$ (Check!). Clearing the positive denominators turns $\Pr(yes|+ve) \geq .999$ into $.001\Pr(yes, +ve) \geq .999\Pr(no, +ve)$ and $\Pr(no|-ve) \geq .7$ into $.3\Pr(no, -ve) \geq .7\Pr(yes, -ve)$, that is

$$\text{(C1)} \qquad .12987\,\delta_1 + .00087\,\delta_2 \;\leq\; -.0005157,$$
$$\text{(C2)} \qquad .039\,\delta_1 + .609\,\delta_2 \;\leq\; -.02229,$$

which are Inequality 16.17 of Corollary 10 with the sides exchanged. Plotting δ_1 horizontally, (C1) is the near-vertical line of slope -149.3 through $(-.003971, 0)$ and $(0, -.592759)$, admissible to its left, and (C2) the near-horizontal line of slope $-.0640$ through $(-.571538, 0)$ and $(0, -.036601)$, admissible below; keeping the rates in $[0, 1]$ adds $\delta_1 \geq -.01$ and $\delta_2 \geq -.10$, so the solution space is the quadrilateral with corners

$$(-.010000,\, -.100000), \quad (-.010000,\, -.035961),$$
$$(-.003727,\, -.036362), \quad (-.003301,\, -.100000).$$

Its vertex $(-.0037273, -.0363623)$, where (C1) and (C2) hold with equality, is the point asked for; there $\theta_{+ve|no} = 19/3029 = .0062727$ and $\theta_{-ve|yes} = 430/6757 = .0636377$, so the false-positive rate falls from 1% to 0.63% and the false-negative rate from 10% to 6.36%, and

$$\Pr(yes|+ve) = \frac{.87 \times .9363623}{.87 \times .9363623 + .13 \times .0062727} = .999,$$
$$\Pr(no|-ve) = \frac{.13 \times .9937273}{.13 \times .9937273 + .87 \times .0636377} = .700.$$

Problem 16.12. Consider a simplified version of the fire network in Figure 16.3 where we are only interested in the variables Fire, Tampering, and Alarm. That is, the network consists of two root nodes F (Fire) and T (Tampering), both of which are parents of the node A (Alarm); all three variables are binary. Its CPTs, taken from Figure 16.3, are

F	θ_F
true	.01
false	.99

T	θ_T
true	.02
false	.98

F	T	A	$\theta^0_{A\|F,T}$
true	true	true	.50
true	false	true	.99
false	true	true	.85
false	false	true	.0001

(redundant CPT rows, for A false, are omitted). Compute the sensitivity function for query $\Pr(F|A)$ in terms of parameter $\theta_{A|F,T}$, then again for the same query in terms of parameter $\theta_{A|\neg F,\neg T}$, and plot the functions. From the plots, what would the effects on the query value be if we apply a small absolute change in each of the two parameters? Also, compute the sensitivity functions for the query $\Pr(A|F)$ in terms of these two parameters. What is the difference between these two sensitivity functions and the previous two?

Solution. By Theorem 16.4 each of $\Pr(F, A)$, $\Pr(A)$, $\Pr(A, F)$ and $\Pr(F)$ is a linear function of the

parameter being varied, so each sensitivity function is a ratio of two linear functions,

$$\Pr(\alpha|\beta) = \frac{\mu^{\alpha,\beta}_{x|u}\,\theta_{x|u} + \nu^{\alpha,\beta}_{x|u}}{\mu^{\beta}_{x|u}\,\theta_{x|u} + \nu^{\beta}_{x|u}}.$$

The coefficients come out directly, the network being small enough to eliminate by hand. Because F and T are independent roots,

$$\Pr(f,t) = .0002, \quad \Pr(f,\bar{t}) = .0098, \quad \Pr(\bar{f},t) = .0198, \quad \Pr(\bar{f},\bar{t}) = .9702.$$

(1) Write $\theta = \theta_{A|F,T}$, current value .50. Summing out T,

$$\begin{aligned}\Pr(F,A) &= .0002\,\theta + .0098 \times .99 = .0002\,\theta + .009702,\\ \Pr(\neg F,A) &= .0198 \times .85 + .9702 \times .0001 = .01683 + .00009702 = .01692702,\end{aligned}$$

the second free of θ, so $\Pr(A) = .0002\,\theta + .02662902$ and

$$\Pr(F|A) \;=\; \frac{.0002\,\theta + .009702}{.0002\,\theta + .02662902}.$$

Some values of this function:

$\theta_{A\|F,T}$	$\Pr(F \mid A)$
.00	.364339
.25	.365531
.50	.366718
.75	.367900
1.00	.369078

Over $[0,1]$ the plot is a rise so gentle as to be indistinguishable from a horizontal line at height $\approx .367$, total variation $.369078 - .364339 = .004739$, with derivative at the current value

$$\left.\frac{d}{d\theta}\Pr(F|A)\right|_{\theta=.5} = \frac{.0002\,(.02662902 - .009702)}{(.5 \times .0002 + .02662902)^2} = .00474.$$

(2) Write $\theta' = \theta_{A|\neg F,\neg T}$, current value .0001. Now

$$\begin{aligned}\Pr(F,A) &= .0002 \times .5 + .0098 \times .99 = .009802,\\ \Pr(\neg F,A) &= .0198 \times .85 + .9702\,\theta' = .01683 + .9702\,\theta',\end{aligned}$$

so $\Pr(F,A)$ is constant in θ', $\Pr(A) = .026632 + .9702\,\theta'$ and

$$\Pr(F|A) \;=\; \frac{.009802}{.026632 + .9702\,\theta'}.$$

Some values:

$\theta_{A\|\neg F,\neg T}$	$\Pr(F \mid A)$
.0000	.368053
.0001	.366718
.0010	.355117
.0100	.269775
.1000	.079271
1.0000	.009833

Over $[0,1]$ this is a steeply decaying hyperbola, down a quarter of its height by $\theta' = .01$, to .0444 at $\theta' = .2$, then flat near zero; over the narrow range $[0,.02]$, as in the two-panel style of Figure 16.4, it plunges convexly from .368 through .270 at $\theta' = .01$ to .213. Its derivative at the current value is

$$\left.\frac{d}{d\theta'}\Pr(F|A)\right|_{\theta'=.0001} = \frac{-.009802 \times .9702}{(.026632 + .9702 \times .0001)^2} = -13.31.$$

Effect of a small absolute change. A change of $+.01$ in $\theta_{A|F,T}$ moves the query from .366718 to .366765, about 5×10^{-5}, and no change at all to it, even to 0 or 1, can leave $[.3643, .3691]$; the same $+.01$ in $\theta_{A|\neg F,\neg T}$ moves the query from .366718 to .269056, a drop of .098, some two thousand times larger. The coefficients say why: $\theta_{A|F,T}$ is multiplied by the tiny prior mass $\Pr(f,t) = .0002$, $\theta_{A|\neg F,\neg T}$ by the dominant $\Pr(\bar{f},\bar{t}) = .9702$.
(3) and (4) For $\Pr(A|F)$: since F and T are roots and A a leaf, $\Pr(F) = .01$ regardless of the CPT of A, and summing out T,

$$\Pr(A|F) = .02\,\theta_{A|F,T} + .98 \times .99 = .02\,\theta_{A|F,T} + .9702,$$

a line of slope .02, value .9802 at the current parameter and range $[.9702, .9902]$ over $\theta_{A|F,T} \in [0,1]$. The other parameter $\theta_{A|\neg F,\neg T}$ appears only in instantiations containing $\neg F$, so in neither $\Pr(A,F)$ nor $\Pr(F)$, and

$$\Pr(A|F) = .9802,$$

a constant, plotted as a horizontal line.
The difference: the first two functions are genuine ratios of linear functions, hence hyperbolas, because the parameter enters both the numerator $\Pr(F,A)$ and the denominator $\Pr(A)$, so $\mu^{\beta}_{x|u} \neq 0$ with $\beta = A$. The last two are linear, one constant, because for $\Pr(A|F)$ the conditioning event is $\beta = F$ and $\Pr(F) = .01$ is free of every parameter in the CPT of A, so $\mu^{F}_{x|u} = 0$ and $(\mu^{\alpha,\beta}\theta + \nu^{\alpha,\beta})/(\mu^{\beta}\theta + \nu^{\beta})$ degenerates to a linear function: evidence on an ancestor of the changed family leaves the denominator alone, evidence on a descendant does not.

Problem 16.13. Assume that instead of using the proportional scheme of Definition 16.2, we distribute the parameter change equally among co-varying parameters. In this case, what is the new value of $\theta_{x^\star|u}$? What problem may this scheme face when distributing the parameter change?
Recall Definition 16.2. Let $\Theta^0_{X|\mathbf{U}}$ be an initial CPT for variable X and suppose we want to change a single parameter value $\theta^0_{x|u}$ to the new value $\theta_{x|u}$. The proportional scheme changes the co-varying parameters, for all $x^\star \neq x$, by

$$\theta_{x^\star|u} = \rho(x,x^\star,u)\,(1-\theta_{x|u}), \qquad \rho(x,x^\star,u) = \begin{cases} \dfrac{\theta^0_{x^\star|u}}{1-\theta^0_{x|u}} & \text{if } \theta^0_{x|u} \neq 1, \\ \dfrac{1}{|X|-1} & \text{if } \theta^0_{x|u} = 1, \end{cases}$$

where $|X|$ is the number of values of X.

Solution. The equal scheme sets, for every $x^\star \neq x$ and $\delta_{x|u} = \theta_{x|u} - \theta^0_{x|u}$,

$$\theta_{x^\star|u} \;=\; \theta^0_{x^\star|u} - \frac{\delta_{x|u}}{|X|-1} \;=\; \theta^0_{x^\star|u} - \frac{\theta_{x|u} - \theta^0_{x|u}}{|X|-1},$$

splitting the $-\delta_{x|u}$ that normalization $\sum_{x'} \theta_{x'|u} = 1$ forces onto the $|X|-1$ co-varying parameters into equal shares; these sum to $1-\theta_{x|u}$ (Check!). It is a common additive shift where Definition 16.2 applies the common multiplicative rescaling $\theta_{x^\star|u} = \theta^0_{x^\star|u}(1-\theta_{x|u})/(1-\theta^0_{x|u})$, so the two agree for binary X and differ only when $|X| \geq 3$.
The problem is that the equal scheme can leave the probability simplex: the subtracted amount is the same for every co-varying parameter while the parameters themselves may differ wildly in size, so a small one can be driven negative. Precisely, an increase ($\delta_{x|u} > 0$) yields a legitimate CPT if and only if

$$\delta_{x|u} \;\leq\; (|X|-1) \cdot \min_{x^\star \neq x} \theta^0_{x^\star|u},$$

so a co-varying parameter of initial value 0 blocks every increase. Decreases are safe, since $\delta_{x|u} < 0$ gives $|\delta_{x|u}| \leq \theta^0_{x|u}$ and hence $\theta_{x^\star|u} \leq \theta^0_{x^\star|u} + \theta^0_{x|u} \leq 1$. Concretely, for the CPT column used in the text,

$\theta^0_{x_1\|u}$	$\theta^0_{x_2\|u}$	$\theta^0_{x_3\|u}$
.6	.3	.1

raising $\theta_{x_1|u}$ from .6 to .9 makes each of the two co-varying parameters give up .15, and the equal scheme yields

$\theta_{x_1\|u}$	$\theta_{x_2\|u}$	$\theta_{x_3\|u}$
.9	.15	−.05

not a probability distribution, whereas the proportional scheme multiplies each co-varying parameter by $(1-.9)/(1-.6) = .25$, giving the legitimate .075 and .025, and can never fail, multiplying nonnegative numbers by a nonnegative factor. Two lesser defects: the equal scheme destroys the ratios $\theta_{x_3|u}/\theta_{x_2|u} = \theta^0_{x_3|u}/\theta^0_{x_2|u}$ that the proportional scheme preserves, and it lifts a zero co-varying parameter whenever $\theta_{x|u}$ decreases, which at $\Pr^0(u) > 0$ changes the support and so sends $D(\Pr^0, \Pr)$ to ∞ by Exercise 16.1, voiding the bounds of Theorem 16.1.

Problem 16.14. Prove that the proportional scheme of Definition 16.2 is the optimal scheme of distributing parameter changes among co-varying parameters in the sense that it minimizes the CD distance $D(\Pr^0, \Pr)$ among all possible schemes.
Here a scheme is any rule that, given an initial CPT $\Theta^0_{X|\mathbf{U}}$, a parent instantiation u, and the new value $\theta_{x|u}$ of a single changed parameter $\theta^0_{x|u}$, assigns new values $\theta_{x^\star|u} \geq 0$ to the co-varying parameters $(x^\star \neq x)$ so that $\sum_{x^\star \neq x} \theta_{x^\star|u} = 1 - \theta_{x|u}$. All other CPTs, and all other columns of the CPT of X, are left unchanged.

Solution. Every scheme's co-varying ratios must average to the scheme-independent value $s = (1 - \theta_{x|u})/(1 - \theta^0_{x|u})$, so none can beat the proportional scheme, which puts every ratio at s. Only the column $\Theta_{X|u}$ is touched, so if $\Pr^0(u) = 0$ then $D(\Pr^0, \Pr) = 0$ for every scheme; and if $\Pr^0(u) > 0$, Theorem 16.2 and (16.3) give

$$D(\Pr^0, \Pr) \;=\; \ln \max_{x'} \frac{\theta_{x'|u}}{\theta^0_{x'|u}} \;-\; \ln \min_{x'} \frac{\theta_{x'|u}}{\theta^0_{x'|u}},$$

the extrema over all values x' of X, with $0/0 = 1$ and $c/0 = \infty$ for $c > 0$; the scheme enters only through these $|X|$ ratios. The degenerate cases tie, the proportional scheme attaining the common minimum:
(i) $\theta^0_{x|u} = 0 < \theta_{x|u}$ the ratio at x is already ∞, so $D = \infty$ whatever the scheme does.
(ii) $\theta^0_{x|u} = \theta_{x|u} = 0$ nothing is asked of the co-varying parameters beyond their original sum, and Definition 16.2 leaves each alone, giving $D = 0$.
(iii) $\theta_{x|u} = 1$ every scheme must zero all co-varying parameters, so there is only one scheme.
(iv) $\theta^0_{x|u} = 1 > \theta_{x|u}$ all co-varying parameters start at 0 and some must be raised, so $D = \infty$ for every scheme.
So assume $0 < \theta^0_{x|u} < 1$ and $0 \leq \theta_{x|u} < 1$, and put

$$r \;=\; \frac{\theta_{x|u}}{\theta^0_{x|u}}, \qquad s \;=\; \frac{1 - \theta_{x|u}}{1 - \theta^0_{x|u}} \;>\; 0.$$

Neither depends on the scheme, $\theta_{x|u}$ being given and the co-varying parameters summing to $1 - \theta_{x|u}$ under any scheme, and $\theta^0_{x|u} r + (1 - \theta^0_{x|u}) s = \theta_{x|u} + (1 - \theta_{x|u}) = 1$ exhibits 1 as a convex combination of r and s, giving the sandwich $\min(r, s) \leq 1 \leq \max(r, s)$.
Definition 16.2 sets $\theta^P_{x^\star|u} = \theta^0_{x^\star|u} \cdot s$, so every co-varying parameter with $\theta^0_{x^\star|u} > 0$ has ratio exactly s, while one with $\theta^0_{x^\star|u} = 0$ stays at 0 and contributes $0/0 = 1$, which by the sandwich moves neither extremum. Hence

$$D^P \;=\; \ln \max(r, s) - \ln \min(r, s) \;=\; |\ln r - \ln s|\,.$$

Let an arbitrary scheme produce values $\theta_{x^\star|u}$. If it makes some $x^\star$ with $\theta^0_{x^\star|u} = 0$ positive then $D = \infty \geq D^P$; so assume it does not, and on $S = \{x^\star \neq x : \theta^0_{x^\star|u} > 0\}$, nonempty as $\sum_{x^\star \neq x} \theta^0_{x^\star|u} =$

$1-\theta^0_{x|u} > 0$, put $\rho_{x^\star} = \theta_{x^\star|u}/\theta^0_{x^\star|u}$. Normalization reads $\sum_{x^\star\in S}\theta^0_{x^\star|u}\rho_{x^\star} = 1-\theta_{x|u}$, so dividing by $1-\theta^0_{x|u} = \sum_{x^\star\in S}\theta^0_{x^\star|u} > 0$,

$$s \;=\; \sum_{x^\star\in S} \frac{\theta^0_{x^\star|u}}{1-\theta^0_{x|u}}\,\rho_{x^\star},$$

a weighted average with nonnegative weights summing to 1, whence $\min_S \rho_{x^\star} \le s \le \max_S \rho_{x^\star}$: no scheme can pull all its co-varying ratios strictly closer to 1 than s is. With M and m the maximum and minimum ratio over the whole column,

$$M \ge \max\left(r,\ \max_{x^\star\in S}\rho_{x^\star}\right) \ge \max(r,s),$$
$$m \le \min\left(r,\ \min_{x^\star\in S}\rho_{x^\star}\right) \le \min(r,s),$$

so that, ln being increasing,

$$D(\Pr^0,\Pr) = \ln M - \ln m \;\ge\; \ln\max(r,s) - \ln\min(r,s) \;=\; D^P.$$

The proportional scheme is itself a scheme and attains D^P, so $D^P = \min_{\text{schemes}} D(\Pr^0,\Pr)$.

15.3 Exercises 16.15–16.17

Problem 16.15. Consider the network in Figure 16.3, reproduced here. It has six binary variables — Fire (F), Tampering (T), Smoke (S), Alarm (A), Leaving (L) and Report (R) — and the edges

$$F \to S, \quad F \to A, \quad T \to A, \quad A \to L, \quad L \to R.$$

Its CPTs are (redundant rows omitted, so each row below states the probability of the value shown):

F	θ_F
true	.01

T	θ_T
true	.02

F	S	$\theta_{S\|F}$
true	true	.9
false	true	.01

F	T	A	$\theta_{A\|F,T}$
true	true	true	.5
true	false	true	.99
false	true	true	.85
false	false	true	.0001

A	L	$\theta_{L\|A}$
true	true	.88
false	true	.001

L	R	$\theta_{R\|L}$
true	true	.75
false	true	.01

The probability of having a fire given that the alarm has triggered is $\Pr(f \mid a) = .3667$. We now wish to install a smoke detector that responds only to whether smoke is present, such that when both the alarm and the smoke detector trigger, the probability of fire is at least .8. How can we use sensitivity analysis to find the required reliability of the smoke detector? Specify the three elements of the sensitivity analysis process: the Bayesian network that models this scenario, the query constraint you wish to satisfy, and the parameters you are allowed to change. Hint: You may recall how sensors are modeled as soft evidence in Chapter 3.

Solution.

(1) The network a smoke detector is the noisy sensor of Section 3.6.4, a binary variable whose only parent is the event it senses, here S. Take N^0 to be Figure 16.3 augmented by a node D and the edge $S \to D$, all six original CPTs unchanged, with $\Theta_{D|S}$ the sensor's error profile

$$f_p \;=\; \Pr(d \mid \neg s) \;=\; \theta_{d|\neg s}, \qquad f_n \;=\; \Pr(\neg d \mid s) \;=\; 1 - \theta_{d|s}.$$

Start it vacuous, $\theta_{d|s} = \theta_{d|\neg s} = 1$: a detector that always triggers has Bayes factor 1, so $\Pr^0(f \mid a, d) = \Pr^0(f \mid a) = .3667$ as quoted, and only $\theta_{d|\neg s}$ is left free, so Corollary 9 applies verbatim.

(2) The query constraint with evidence $e = a, d$ and event $y = f$, the type-(16.9) constraint $\Pr(f \mid a, d) \geq \kappa$, $\kappa = .8$.

(3) The changeable parameters only the sensor CPT, $\theta_{d|s} = 1 - f_n$ and $\theta_{d|\neg s} = f_p$ — one is buying a detector, not re-engineering fires. Two parameters in one CPT is the multiple-parameter setting of Section 16.3.2; pinning $\theta_{d|s} = 1$ reduces it to Section 16.3.1.

Since D is binary, the only co-varying parameter is $\theta_{\neg d|\neg s}$ and $\rho = 1$ in (16.5); both query events $\alpha = f, a, d$ and $\alpha = a, d$ mention d, so $\partial \Pr(\alpha)/\partial\theta_{\neg d|\neg s} = 0$, and $\partial \Pr(\alpha)/\partial\theta_{d|\neg s} = \Pr^0(\alpha, d, \neg s)/\theta_{d|\neg s}$ at $\theta_{d|\neg s} = 1$ makes the constants of (16.5)

$$\mu^{f,a,d}_{d|\neg s} = \Pr^0(f, a, \neg s), \qquad \mu^{a,d}_{d|\neg s} = \Pr^0(a, \neg s).$$

Inference in Figure 16.3 gives

$$\begin{aligned} \Pr^0(a) &= .026729, & \Pr^0(f, a) &= .009802, \\ \Pr^0(a, \neg s) &= .017738, & \Pr^0(f, a, \neg s) &= .00098020. \end{aligned}$$

In N^0 the detector always fires, so $\Pr^0(f, a, d) = \Pr^0(f, a)$ and $\Pr^0(a, d) = \Pr^0(a)$, with ratio .3667 as claimed. Substituting into (16.13) with $\kappa = .8$:

$$\begin{aligned} \Pr^0(f, a, d) - \kappa \Pr^0(a, d) \;&\geq\; \delta_{d|\neg s}\big(-\mu^{f,a,d}_{d|\neg s} + \kappa\, \mu^{a,d}_{d|\neg s}\big) \\ .009802 - .8(.026729) \;&\geq\; \delta_{d|\neg s}\big(-.00098020 + .8(.017738)\big) \\ -.0115812 \;&\geq\; \delta_{d|\neg s}\,(.0132102). \end{aligned}$$

Hence $\delta_{d|\neg s} \leq -.87669$, and since the initial value is 1 the new value must satisfy

$$\theta_{d|\neg s} \;=\; f_p \;\leq\; .12331.$$

So a detector that never misses smoke ($f_n = 0$) needs a false positive rate of at most 12.3%.

Method (2): allow f_n to move too. The only path from D to A is $D \leftarrow S \leftarrow F \to A$, with F divergent, so F d-separates D from A and S from A; hence $\Pr(s \mid f, a) = .9$, $\Pr(s \mid \neg f, a) = .01$, and D enters the odds of F only through $\Pr(d \mid f, a) = \Pr(d \mid f)$ and $\Pr(d \mid \neg f, a) = \Pr(d \mid \neg f)$. With $q = \Pr(f \mid a) = .3667$, sensor likelihoods $\Pr(d \mid s) = 1 - f_n$, $\Pr(d \mid \neg s) = f_p$, and Bayes rule in odds form,

$$O(f \mid a, d) \;=\; O(f \mid a) \cdot \frac{\Pr(d \mid f, a)}{\Pr(d \mid \neg f, a)} \;=\; \frac{q}{1-q} \cdot \frac{.9(1 - f_n) + .1 f_p}{.01(1 - f_n) + .99 f_p}.$$

Requiring $\Pr(f \mid a, d) \geq .8$ is requiring $O(f \mid a, d) \geq 4$; dividing through by f_p and writing $k^+ = (1 - f_n)/f_p$ for the Bayes factor of a positive reading (Section 3.6.4), this is

$$\frac{q}{1-q} \cdot \frac{.9k^+ + .1}{.01k^+ + .99} \;\geq\; 4.$$

With $q/(1-q) = .57907$ this reads $(.9k^+ + .1)/(.01k^+ + .99) \geq 6.9076$, that is,

$$.83092\, k^+ \;\geq\; 6.73850, \qquad \text{so} \qquad k^+ \;\geq\; 8.1096.$$

The required reliability is thus a constraint on the ratio alone,

$$\frac{1 - f_n}{f_p} \;\geq\; 8.1096, \qquad \text{equivalently} \qquad f_p \;\leq\; \frac{1 - f_n}{8.1096},$$

as Section 3.6.4 predicts. So $f_n = 0$ requires $f_p \leq .1233$, matching the Corollary 9 computation; $f_n = .05$ requires $f_p \leq .1171$; $f_n = .10$ requires $f_p \leq .1110$.

Problem 16.16. Let N^0 and N be Bayesian networks where network N is obtained from N^0 by changing the CPTs of variables X and Y from $\Theta^0_{X|U_X}$ to $\Theta_{X|U_X}$ and from $\Theta^0_{Y|U_Y}$ to $\Theta_{Y|U_Y}$, respectively — that is, changing parameter value $\theta^0_{x|u_X}$ to some new value $\theta_{x|u_X}$ for every x and u_X, and $\theta^0_{y|u_Y}$ to some new value $\theta_{y|u_Y}$ for every y and u_Y. Let $\Pr^0$ and $\Pr$ be the distributions induced by networks N^0 and N, respectively, and also assume that $\Pr^0(u_X) > 0$ and $\Pr^0(u_Y) > 0$ for every u_X and u_Y. Prove that if the two families X, U_X and Y, U_Y are disjoint, that is, they do not share any variables, then

$$D(\Pr^0, \Pr) \;=\; D(\Theta^0_{X|U_X}, \Theta_{X|U_X}) \;+\; D(\Theta^0_{Y|U_Y}, \Theta_{Y|U_Y}).$$

Also prove that if the two families are not disjoint, then the sum of the distances between the CPTs is an upper bound on $D(\Pr^0, \Pr)$. Here the CD distance between two CPTs is the CD distance of Definition 16.1 applied entrywise, that is,

$$D(\Theta^0_{X|U_X}, \Theta_{X|U_X}) = \ln \max_{x,u_X} \frac{\theta_{x|u_X}}{\theta^0_{x|u_X}} - \ln \min_{x,u_X} \frac{\theta_{x|u_X}}{\theta^0_{x|u_X}},$$

with the conventions $0/0 = 1$ and $\infty/\infty = 1$ of Definition 16.1.

Solution. Both claims follow from the pointwise bound $(*)$ below, which is an equality exactly when the extreme entries of the two CPTs are realizable in one instantiation. Let z range over complete instantiations of the network variables Z, write x, u_X and y, u_Y for the values z assigns to the two families, and abbreviate

$$r_X(x, u_X) = \frac{\theta_{x|u_X}}{\theta^0_{x|u_X}}, \qquad r_Y(y, u_Y) = \frac{\theta_{y|u_Y}}{\theta^0_{y|u_Y}},$$

and put

$$M_X = \max_{x,u_X} r_X(x, u_X), \qquad m_X = \min_{x,u_X} r_X(x, u_X),$$
$$M_Y = \max_{y,u_Y} r_Y(y, u_Y), \qquad m_Y = \min_{y,u_Y} r_Y(y, u_Y),$$

so that $D(\Theta^0_{X|U_X}, \Theta_{X|U_X}) = \ln M_X - \ln m_X$ and likewise for Y.

By the chain rule $\Pr^0(z)$ and $\Pr(z)$ are products of one parameter per network variable, differing only in the factors from X and Y, so with $c(z)$ the product of the parameters from all other variables (identical in both networks),

$$\Pr^0(z) = \theta^0_{x|u_X}\, \theta^0_{y|u_Y}\, c(z),$$
$$\Pr(z) = \theta_{x|u_X}\, \theta_{y|u_Y}\, c(z).$$

valid disjoint or not, X and Y being distinct variables, so exactly one factor comes from each changed CPT.

Each CPT ratio straddles 1. Fix u_X; since $\sum_x \theta^0_{x|u_X} = \sum_x \theta_{x|u_X} = 1$, neither $\theta_{x|u_X} > \theta^0_{x|u_X}$ for every x nor the reverse is possible, so some x has $\theta_{x|u_X} \le \theta^0_{x|u_X}$ and some the reverse. The first gives $r_X(x, u_X) \le 1$ (clear when $\theta^0_{x|u_X} > 0$, and when $\theta^0_{x|u_X} = 0$ it forces $\theta_{x|u_X} = 0$ and $r_X = 0/0 = 1$), the second symmetrically an entry with $r_X \ge 1$. Therefore

$$m_X \le 1 \le M_X, \qquad \text{and likewise} \qquad m_Y \le 1 \le M_Y.$$

The upper bound, disjoint or not, rests on the claim that for every z,

$$m_X\, m_Y \;\le\; \frac{\Pr(z)}{\Pr^0(z)} \;\le\; M_X\, M_Y. \qquad (*)$$

Four cases.

(i) $\Pr^0(z) > 0$ then $c(z), \theta^0_{x|u_X}, \theta^0_{y|u_Y} > 0$, so $c(z)$ cancels and $\Pr(z)/\Pr^0(z) = r_X(x, u_X)\, r_Y(y, u_Y)$, a product of nonnegative ratios lying between $m_X m_Y$ and $M_X M_Y$.

(ii) $\Pr^0(z) = \Pr(z) = 0$ the ratio is $0/0 = 1$, inside the interval by the straddle.
(iii) $\Pr^0(z) = 0 < \Pr(z)$ the ratio is ∞; here $c(z) > 0$ and $\theta_{x|u_X}\theta_{y|u_Y} > 0 = \theta^0_{x|u_X}\theta^0_{y|u_Y}$, so one changed parameter has $\theta > 0 = \theta^0$, entrywise ratio ∞, and the other factor is ≥ 1, giving $M_X M_Y = \infty$; the lower bound holds as $m_X m_Y \leq 1$.
(iv) $\Pr(z) = 0 < \Pr^0(z)$ the ratio is 0; symmetrically some changed parameter has $\theta = 0 < \theta^0$, entrywise ratio 0, and the other minimum is ≤ 1 and finite, giving $m_X m_Y = 0$; the upper bound holds as $M_X M_Y \geq 1$.

Taking extrema of $(*)$ over z and applying Definition 16.1,

$$\begin{aligned} D(\Pr^0, \Pr) &= \ln \max_z \frac{\Pr(z)}{\Pr^0(z)} - \ln \min_z \frac{\Pr(z)}{\Pr^0(z)} \\ &\leq \ln(M_X M_Y) - \ln(m_X m_Y) \\ &= (\ln M_X - \ln m_X) + (\ln M_Y - \ln m_Y) \\ &= D(\Theta^0_{X|U_X}, \Theta_{X|U_X}) + D(\Theta^0_{Y|U_Y}, \Theta_{Y|U_Y}), \end{aligned}$$

the required upper bound; splitting the logarithms is legitimate since $M_X, M_Y \geq 1$ and $m_X, m_Y \leq 1$ bar any $\infty - \infty$.

For equality in the disjoint case it remains to attain the two extremes of $(*)$. Choose entries $(x^\star, u^\star_X)$ attaining M_X and $(y^\star, u^\star_Y)$ attaining M_Y; disjointness of the families is exactly what makes $x^\star, u^\star_X, y^\star, u^\star_Y$ a consistent instantiation. Under the nondegeneracy hypothesis

$$\Pr^0(x, u_X, y, u_Y) > 0 \quad \text{for every instantiation of } X, U_X, Y, U_Y, \tag{$\dagger$}$$

some complete $z^\star$ extends $x^\star, u^\star_X, y^\star, u^\star_Y$ with $\Pr^0(z^\star) > 0$, and case (i) gives

$$\frac{\Pr(z^\star)}{\Pr^0(z^\star)} = r_X(x^\star, u^\star_X)\, r_Y(y^\star, u^\star_Y) = M_X M_Y.$$

The minimizing entries likewise produce a $z_\star$ with ratio $m_X m_Y$, so the extrema over z are $M_X M_Y$ and $m_X m_Y$ and the displayed chain becomes a chain of equalities:

$$D(\Pr^0, \Pr) = D(\Theta^0_{X|U_X}, \Theta_{X|U_X}) + D(\Theta^0_{Y|U_Y}, \Theta_{Y|U_Y}).$$

The printed hypotheses $\Pr^0(u_X) > 0$ and $\Pr^0(u_Y) > 0$ do not suffice for this last step (nor does the joint reading $\Pr^0(u_X, u_Y) > 0$), so we prove the equality under $(\dagger)$, which holds whenever N^0 is strictly positive and is the two-family analogue of the $\Pr^0(u) > 0$ of Theorem 16.2.

Problem 16.17. Let N^0 and N be Bayesian networks where network N is obtained from N^0 by changing the CPTs of variables $X_1, \ldots, X_m$ from $\Theta^0_{X_i|U_{X_i}}$ to $\Theta_{X_i|U_{X_i}}$ (i.e., changing parameter value $\theta^0_{x_i|u_{X_i}}$ to some new value $\theta_{x_i|u_{X_i}}$ for every x_i and u_{X_i}). Let $\Pr^0$ and $\Pr$ be the distributions induced by networks N^0 and N, respectively, and also assume that $\Pr^0(u_{X_i}) > 0$ for every u_{X_i}. Devise a procedure that computes $D(\Pr^0, \Pr)$. Hint: This procedure can be similar to the one used for computing the MPE probability.

Solution. Run the MPE algorithms of Chapter 10 twice on the entrywise ratio tables, once maximizing and once minimizing: the CD distance of Definition 16.1 asks for the extremes of a quantity that factorizes over the families of the network, which is exactly what those algorithms compute. Let z range over complete instantiations of the network variables Z, and for each changed X_i define the ratio table

$$f_i(X_i, U_{X_i}), \qquad f_i(x_i, u_{X_i}) \stackrel{\text{def}}{=} \frac{\theta_{x_i|u_{X_i}}}{\theta^0_{x_i|u_{X_i}}},$$

a factor over exactly the family X_i, U_{X_i}. By the chain rule $\Pr^0(z)$ and $\Pr(z)$ are products of one

parameter per network variable differing only in the factors from $X_1, \ldots, X_m$, so with $c(z)$ the product of the parameters from the remaining variables (identical in both networks), every z with $\Pr^0(z) > 0$ has

$$\frac{\Pr(z)}{\Pr^0(z)} = \frac{c(z)\prod_{i=1}^m \theta_{x_i|u_{X_i}}}{c(z)\prod_{i=1}^m \theta^0_{x_i|u_{X_i}}} = \prod_{i=1}^m f_i(x_i, u_{X_i}),$$

where x_i, u_{X_i} are the values assigned by z to X_i, U_{X_i}. Hence, by Definition 16.1,

$$D(\Pr^0, \Pr) = \ln\Big(\max_z \prod_{i=1}^m f_i\Big) - \ln\Big(\min_z \prod_{i=1}^m f_i\Big). \qquad (*)$$

This is the MPE computation $\mathrm{MPE}_P = \max_z \prod_V \theta_{v|u_V}$ with each CPT replaced by its ratio table (and by the constant 1 for unchanged variables); nothing in Chapter 10 uses normalization of the tables, so that machinery applies verbatim. The procedure:

1. Build the factor set $\mathcal{F} = \{f_1, \ldots, f_m\}$ by dividing the new CPTs entrywise by the old ones, using $0/0 = 1$. Unchanged variables contribute nothing (equivalently, the constant factor 1).
2. Choose an elimination order π of all network variables Z. Since every f_i has scope equal to a family of the network, the interaction graph of $\mathcal{F}$ is a subgraph of the moral graph of N^0; so any order of width w for the network is an order of width at most w here.
3. Run VE_MPE (Algorithm 27) on $\mathcal{F}$ with order π: eliminate the variables one at a time, multiplying all factors that mention the variable and then maximizing it out (Definition 10.1). Eliminating all variables leaves a trivial factor whose value is $R_{\max} = \max_z \prod_i f_i$.
4. Run the same elimination again with maximization replaced by minimization, giving $R_{\min} = \min_z \prod_i f_i$.
5. Return $D(\Pr^0, \Pr) = \ln R_{\max} - \ln R_{\min}$.

The min pass is legitimate because the correctness of VE_MPE rests on Theorem 10.1, $\max_X f_1 f_2 = f_1 \max_X f_2$ when X occurs only in f_2, and the same holds for minimization of nonnegative factors since $\min(ab, ac) = a\min(b, c)$ for $a \geq 0$; our f_i are nonnegative, so the identical induction gives $\min_z \prod_i f_i$. Alternatively, leave the algorithm alone and call it on the reciprocal tables $1/f_i$, using

$$\min_z \prod_{i=1}^m f_i \;=\; \Big(\max_z \prod_{i=1}^m \frac{1}{f_i}\Big)^{-1},$$

valid when all entries $\theta_{x_i|u_{X_i}}$, $\theta^0_{x_i|u_{X_i}}$ are positive. This is the form the hint points at: two MPE computations, one on Θ/Θ^0 and one on Θ^0/Θ. Each pass eliminates over factors whose scopes are network families, so each costs $O(n\exp(w))$ time and space, the cost of one MPE computation, and the max-product jointree algorithm may be substituted with the ratio tables assigned to clusters containing their families.

Two points of care, both about instantiations of probability zero.

(i) an entry with $\theta^0_{x_i|u_{X_i}} = 0 < \theta_{x_i|u_{X_i}}$ or $\theta_{x_i|u_{X_i}} = 0 < \theta^0_{x_i|u_{X_i}}$ puts ∞ or 0 in its ratio table. Running both passes in the extended nonnegative reals (no $0 \cdot \infty$ arises, no factor being both) returns $R_{\max} = \infty$ or $R_{\min} = 0$, hence $D(\Pr^0, \Pr) = \infty$, right whenever the offending entry is realizable, as then the supports differ and Exercise 16.1 applies.

(ii) $(*)$ ranges over all z, whereas $\Pr(z)/\Pr^0(z) = \prod_i f_i(z)$ only for z with $c(z) > 0$; instantiations killed by an unchanged parameter have $\Pr(z) = \Pr^0(z) = 0$, true ratio 1. So $(*)$ is exact under the standing assumption that $\Pr^0$ is strictly positive, and an upper bound in general. To restore exactness against structural zeros, attach to every unchanged variable V the indicator table

$$I_V(v, u_V) = \begin{cases} 1 & \text{if } \theta^0_{v|u_V} > 0 \\ 0 & \text{otherwise,} \end{cases}$$

and run the max pass on $\mathcal{F} \cup \{I_V : V \text{ unchanged}\}$, reading $0 \cdot \infty$ as 0 so masking wins; for the min pass use the tables that are 1 where $\theta^0_{v|u_V} > 0$ and ∞ elsewhere, reading $\infty \cdot 0$ as ∞, so forbidden instantiations are excluded rather than driving the minimum to 0. Scopes are unchanged, so the complexity stays $O(n\exp(w))$. The value 1 contributed by the excluded instantiations never falls

outside the resulting interval: $\sum_z \Pr(z) = \sum_z \Pr^0(z) = 1$ forces some surviving z with $\Pr(z) \le \Pr^0(z)$ and some with $\Pr(z) \ge \Pr^0(z)$ when the supports agree, and when they do not both this and the passes report ∞ by Exercise 16.1.

16 Learning: The Maximum Likelihood Approach

Contents

16.1 Exercises 17.1–17.7 288
16.2 Exercises 17.8–17.14 293
16.3 Exercises 17.15–17.20 300

16.1 Exercises 17.1–17.7

Problem 17.1. Consider a Bayesian network structure with the following edges $A \to B$, $A \to C$, and $A \to D$ (so A is a root and B, C, D are three leaves, each with the single parent A; all four variables are binary with values T and F). Compute the ML parameter estimates for this structure given the following data set:

Case	A	B	C	D
1	T	F	F	F
2	T	F	F	T
3	F	F	T	F
4	T	T	F	T
5	F	F	T	T
6	F	T	T	F
7	F	T	T	T
8	T	F	F	T
9	F	F	T	F
10	T	T	T	T

Solution. The data set is complete, so by Theorem 17.1 the ML estimates are the empirical conditional frequencies of Equation 17.1,

$$\theta^{ml}_{x|u} \;=\; Pr_{\mathcal{D}}(x|u) \;=\; \frac{\mathcal{D}\#(xu)}{\mathcal{D}\#(u)},$$

with $\mathcal{D}\#(\alpha)$ the number of cases satisfying α; they are unique here because both parent instantiations occur, so $\mathcal{D}\#(u) > 0$ throughout. Cases $1, 2, 4, 8, 10$ have $A = T$ and cases $3, 5, 6, 7, 9$ have $A = F$, so $\mathcal{D}\#(a) = \mathcal{D}\#(\bar{a}) = 5$ out of $N = 10$; within $A = T$ the counts are $\mathcal{D}\#(ba) = 2$, $\mathcal{D}\#(ca) = 1$, $\mathcal{D}\#(da) = 4$, and within $A = F$ they are $\mathcal{D}\#(b\bar{a}) = 2$, $\mathcal{D}\#(c\bar{a}) = 5$, $\mathcal{D}\#(d\bar{a}) = 2$ (Check!). Hence

$$\begin{aligned} \theta^{ml}_{a} = \theta^{ml}_{\bar{a}} &= .5, & \theta^{ml}_{b|a} = \theta^{ml}_{b|\bar{a}} &= \tfrac{2}{5} = .4, \\ \theta^{ml}_{c|a} &= \tfrac{1}{5} = .2, & \theta^{ml}_{c|\bar{a}} &= \tfrac{5}{5} = 1, \\ \theta^{ml}_{d|a} &= \tfrac{4}{5} = .8, & \theta^{ml}_{d|\bar{a}} &= \tfrac{2}{5} = .4, \end{aligned}$$

which collect into the CPTs

A	θ^{ml}_{A}
T	.5
F	.5

A	B	$\theta^{ml}_{B\|A}$
T	T	.4
T	F	.6
F	T	.4
F	F	.6

A	C	$\theta^{ml}_{C\|A}$
T	T	.2
T	F	.8
F	T	1
F	F	0

A	D	$\theta^{ml}_{D\|A}$
T	T	.8
T	F	.2
F	T	.4
F	F	.6

Problem 17.2. Consider a Bayesian network structure with edges $A \to B$ and $B \to C$ (a chain: A is a root, B has the single parent A, and C has the single parent B; all three variables are binary with values T and F). Compute the ML parameter estimates for this structure given the following data set:

Case	A	B	C
1	T	F	F
2	T	F	F
3	F	F	T
4	T	F	F

Are the ML estimates unique for this data set? If not, how many ML estimates do we have in this case?

Solution. The data set is complete, so Theorem 17.1 gives the ML estimates as the empirical conditional frequencies $\theta^{ml}_{x|u} = \mathcal{D}\#(xu)/\mathcal{D}\#(u)$ wherever $\mathcal{D}\#(u) > 0$:

$$\begin{aligned} \theta^{ml}_{a} &= \tfrac{3}{4} = .75, & \theta^{ml}_{\bar a} &= \tfrac{1}{4} = .25, \\ \theta^{ml}_{b|a} = \theta^{ml}_{b|\bar a} &= 0, & \theta^{ml}_{\bar b|a} = \theta^{ml}_{\bar b|\bar a} &= 1, \\ \theta^{ml}_{c|\bar b} &= \tfrac{1}{4} = .25, & \theta^{ml}_{\bar c|\bar b} &= \tfrac{3}{4} = .75, \end{aligned}$$

from $N = 4$, $\mathcal{D}\#(a) = 3$ (cases $1, 2, 4$), $\mathcal{D}\#(\bar a) = 1$, $\mathcal{D}\#(b) = 0$, $\mathcal{D}\#(\bar b) = 4$ and $\mathcal{D}\#(c\bar b) = 1$ (Check!); the row for b is undefined by Equation 17.1 since $\mathcal{D}\#(b) = 0$.
They are not unique. As in Exercise 17.3, $Pr_{\mathcal{D}}(b) = 0$ leaves the likelihood free of $\theta_{c|b}$ and $\theta_{\bar c|b}$: for complete data the chain rule gives $Pr_\theta(d_i) = \prod_{\mathbf{XU}} \theta_{x_i|u_i}$, so

$$L(\theta|\mathcal{D}) \;=\; \prod_{\mathbf{XU}} \prod_{xu} \theta_{x|u}^{\mathcal{D}\#(xu)},$$

in which the exponents $\mathcal{D}\#(cb)$ and $\mathcal{D}\#(\bar c b)$ are 0, while Theorem 17.1 pins every other parameter. The ML parameterizations are therefore

$$\{\theta : \theta_{c|b} = t,\ \theta_{\bar c|b} = 1 - t,\ t \in [0,1],\ \text{all other parameters as above}\},$$

a one-parameter continuum, hence uncountably many. All attain the same value: cases $1, 2, 4$ each have probability $\theta_a \theta_{\bar b|a} \theta_{\bar c|\bar b} = 9/16$ and case 3 has $\theta_{\bar a} \theta_{\bar b|\bar a} \theta_{c|\bar b} = 1/16$, so

$$L(\mathcal{G}|\mathcal{D}) \;=\; \left(\tfrac{9}{16}\right)^3 \cdot \tfrac{1}{16} \;=\; \frac{729}{65536} \;\approx\; 1.11 \times 10^{-2}.$$

Problem 17.3. Let $\mathcal{G}$ be a network structure with families $\mathbf{XU}$ (each family consists of a variable X together with its parents $\mathbf{U}$ in $\mathcal{G}$), let $\mathcal{D}$ be a complete data set, and suppose that $Pr_{\mathcal{D}}(\mathbf{u}) = 0$ for some instantiation $\mathbf{u}$ of parent set $\mathbf{U}$. Show that the likelihood function $LL(\theta|\mathcal{D})$ is independent of parameters $\theta_{x|\mathbf{u}}$ for all x.

Solution. The parameter $\theta_{x|\mathbf{u}}$ enters $LL(\theta|\mathcal{D})$ only through the term $\mathcal{D}\#(x\mathbf{u}) \log \theta_{x|\mathbf{u}}$, whose coefficient $Pr_{\mathcal{D}}(\mathbf{u}) = 0$ forces to vanish.
Completeness of $\mathcal{D}$ makes the chain rule evaluate each case as one parameter per family, $Pr_\theta(d_i) = \prod_{\mathbf{XU}} \theta_{x_i|\mathbf{u}_i}$; taking logarithms and grouping the N cases by family instantiation (each $\log \theta_{x|\mathbf{u}}$ arising from exactly the $\mathcal{D}\#(x\mathbf{u})$ cases with $d_i \models x\mathbf{u}$),

$$LL(\theta|\mathcal{D}) = \sum_{i=1}^{N} \sum_{\mathbf{XU}} \log \theta_{x_i|\mathbf{u}_i} = \sum_{\mathbf{XU}} \sum_{x\mathbf{u}} \mathcal{D}\#(x\mathbf{u})\ \log \theta_{x|\mathbf{u}}.$$

Now $Pr_{\mathcal{D}}(\mathbf{u}) = 0$ means $\mathcal{D}\#(\mathbf{u}) = 0$ (Definition 17.1), and $x\mathbf{u} \models \mathbf{u}$ gives $0 \le \mathcal{D}\#(x\mathbf{u}) \le \mathcal{D}\#(\mathbf{u}) = 0$ for every x. So no term mentions $\theta_{x|\mathbf{u}}$, and replacing the row $\theta_{X|\mathbf{u}}$ by any other distribution over X

leaves $LL(\theta|\mathcal{D})$ unchanged.

Problem 17.4. Let $\mathcal{G}$ be a network structure with families $\mathbf{XU}$ and let $\mathcal{D}$ be a complete data set. Prove the following form for the likelihood of structure $\mathcal{G}$:

$$L(\mathcal{G}|\mathcal{D}) \;=\; \prod_{\mathbf{XU}} \prod_{x\mathbf{u}} \left(\theta^{ml}_{x|\mathbf{u}}\right)^{\mathcal{D}\#(x\mathbf{u})},$$

where $\theta^{ml}_{x|\mathbf{u}}$ is the ML estimate for parameter $\theta_{x|\mathbf{u}}$, and $\mathcal{D}\#(x\mathbf{u})$ is the number of cases in $\mathcal{D}$ satisfying the family instantiation $x\mathbf{u}$.

Solution. Since $L(\mathcal{G}|\mathcal{D}) = L(\theta^{ml}|\mathcal{D})$ by definition, it suffices to prove for an arbitrary parameterization θ of $\mathcal{G}$ that

$$(*) \qquad L(\theta|\mathcal{D}) \;=\; \prod_{\mathbf{XU}} \prod_{x\mathbf{u}} \theta^{\mathcal{D}\#(x\mathbf{u})}_{x|\mathbf{u}}.$$

Completeness of $\mathcal{D}$ makes the chain rule give $Pr_\theta(d_i) = \prod_{\mathbf{XU}} \theta_{x_i|\mathbf{u}_i}$, with $x_i\mathbf{u}_i$ the unique family instantiation compatible with d_i; multiplying over the N cases (Equation 17.2) and grouping, for each family, the cases by the instantiation they select, so that $\theta_{x|\mathbf{u}}$ is contributed by exactly the $\mathcal{D}\#(x\mathbf{u})$ cases with $d_i \models x\mathbf{u}$,

$$L(\theta|\mathcal{D}) = \prod_{\mathbf{XU}} \prod_{i=1}^{N} \theta_{x_i|\mathbf{u}_i} = \prod_{\mathbf{XU}} \prod_{x\mathbf{u}} \theta^{\mathcal{D}\#(x\mathbf{u})}_{x|\mathbf{u}},$$

the regrouping being exact because over complete cases the family instantiations are mutually exclusive and exhaustive, $\sum_{x\mathbf{u}} \mathcal{D}\#(x\mathbf{u}) = N$. Instantiating $(*)$ at $\theta^{ml}_{x|\mathbf{u}} = Pr_{\mathcal{D}}(x|\mathbf{u})$ (Theorem 17.1) gives the claim; where $\mathcal{D}\#(\mathbf{u}) = 0$ leaves $\theta^{ml}_{X|\mathbf{u}}$ undetermined, Exercise 17.3 makes those counts 0 and those factors 1.

Problem 17.5. Consider a Bayesian network with edges $A \to B$ and $A \to C$ (a common cause: A is a root with the two children B and C, and B, C have no other parents; all variables are binary with values T and F), and the parameters θ:

A	θ_A
T	.3
F	.7

A	B	$\theta_{B\|A}$
T	T	.5
T	F	.5
F	T	.8
F	F	.2

A	C	$\theta_{C\|A}$
T	T	.1
T	F	.9
F	T	.5
F	F	.5

Consider the following data set $\mathcal{D}$ (a question mark denotes a missing value):

Case	A	B	C
1	F	?	T
2	F	T	T
3	?	F	T
4	?	T	F
5	T	F	?

What is the expected empirical distribution for data set $\mathcal{D}$ given parameters θ, $Pr_{\mathcal{D},\theta}(.)$?

Solution. By Definition 17.2 with $N = 5$,

$$Pr_{\mathcal{D},\theta}(\alpha) \;=\; \frac{1}{5} \sum_{d_i, c_i \models \alpha} Pr_\theta(c_i | d_i),$$

c_i ranging over the instantiations of the variables missing from case d_i. The chain rule $Pr_\theta(abc) = \theta_a \theta_{b|a} \theta_{c|a}$ gives the joint:

A	B	C	Pr_θ
T	T	T	$(.3)(.5)(.1) = .015$
T	T	F	$(.3)(.5)(.9) = .135$
T	F	T	$(.3)(.5)(.1) = .015$
T	F	F	$(.3)(.5)(.9) = .135$
F	T	T	$(.7)(.8)(.5) = .280$
F	T	F	$(.7)(.8)(.5) = .280$
F	F	T	$(.7)(.2)(.5) = .070$
F	F	F	$(.7)(.2)(.5) = .070$

Normalizing the consistent rows of this table gives each case's completion weights $Pr_\theta(c_i|d_i)$:

- (i) $d_1 = \bar{a}c$, missing B: rows .28, .07 total .35, so $Pr_\theta(b|\bar{a}c) = .8$, $Pr_\theta(\bar{b}|\bar{a}c) = .2$.
- (ii) $d_2 = \bar{a}bc$ is complete: single completion, weight 1.
- (iii) $d_3 = \bar{b}c$, missing A: rows .015, .07 total .085, so $Pr_\theta(a|\bar{b}c) = 3/17$, $Pr_\theta(\bar{a}|\bar{b}c) = 14/17$.
- (iv) $d_4 = b\bar{c}$, missing A: rows .135, .28 total .415, so $Pr_\theta(a|b\bar{c}) = 27/83$, $Pr_\theta(\bar{a}|b\bar{c}) = 56/83$.
- (v) $d_5 = a\bar{b}$, missing C: rows .015, .135 total .15, so $Pr_\theta(c|a\bar{b}) = .1$, $Pr_\theta(\bar{c}|a\bar{b}) = .9$.

Summing each full instantiation's weights and dividing by 5 — $a\bar{b}c$, say, occurs in (iii) and (v), giving $(3/17 + .1)/5 = 47/850$ — yields:

A	B	C	contributions	$Pr_{\mathcal{D},\theta}(.)$
T	T	T	none	0
T	T	F	$d_4 : 27/83$	$27/415 \approx .06506$
T	F	T	$d_3 : 3/17$, $d_5 : .1$	$47/850 \approx .05529$
T	F	F	$d_5 : .9$	.18
F	T	T	$d_1 : .8$, $d_2 : 1$	$1.8/5 = .36$
F	T	F	$d_4 : 56/83$	$56/415 \approx .13494$
F	F	T	$d_1 : .2$, $d_3 : 14/17$	$87/425 \approx .20471$
F	F	F	none	0

Problem 17.6. Consider Exercise 17.5, that is, the Bayesian network with edges $A \to B$ and $A \to C$ whose CPTs are $\theta_a = .3$; $\theta_{b|a} = .5$, $\theta_{b|\bar{a}} = .8$; $\theta_{c|a} = .1$, $\theta_{c|\bar{a}} = .5$, together with the incomplete data set

Case	A	B	C
1	F	?	T
2	F	T	T
3	?	F	T
4	?	T	F
5	T	F	?

and assume that the given CPTs are the initial CPTs θ^0 used by EM. Compute the EM parameter estimates after one iteration of the algorithm.

Solution. By Definition 17.3, $\theta^1_{x|\mathbf{u}} = Pr_{\mathcal{D},\theta^0}(x\mathbf{u}) / Pr_{\mathcal{D},\theta^0}(\mathbf{u})$, read off the expected empirical distribution of Exercise 17.5:

A	B	C	$Pr_{\mathcal{D},\theta^0}(.)$
T	T	T	0
T	T	F	$27/415 \approx .06506$
T	F	T	$47/850 \approx .05529$
T	F	F	.18
F	T	T	.36
F	T	F	$56/415 \approx .13494$
F	F	T	$87/425 \approx .20471$
F	F	F	0

Summing rows gives the marginals

$$\begin{aligned} Pr_{\mathcal{D},\theta^0}(a) &= \tfrac{27}{415} + \tfrac{47}{850} + \tfrac{9}{50} = \tfrac{2119}{7055} \approx .30035, \\ Pr_{\mathcal{D},\theta^0}(ab) &= \tfrac{27}{415}, \qquad Pr_{\mathcal{D},\theta^0}(\bar{a}b) = .36 + \tfrac{56}{415} = \tfrac{1027}{2075}, \\ Pr_{\mathcal{D},\theta^0}(ac) &= \tfrac{47}{850}, \qquad Pr_{\mathcal{D},\theta^0}(\bar{a}c) = .36 + \tfrac{87}{425} = \tfrac{48}{85}, \end{aligned}$$

and dividing by $Pr_{\mathcal{D},\theta^0}(a)$ and $Pr_{\mathcal{D},\theta^0}(\bar{a}) = 4936/7055$,

$$\begin{aligned} \theta^1_{b|a} &= \tfrac{459}{2119} \approx .21661, \qquad \theta^1_{b|\bar{a}} = \tfrac{17459}{24680} \approx .70741, \\ \theta^1_{c|a} &= \tfrac{3901}{21190} \approx .18410, \qquad \theta^1_{c|\bar{a}} = \tfrac{498}{617} \approx .80713. \end{aligned}$$

The estimates after one EM iteration are therefore:

A	θ^1_A
T	.30035
F	.69965

A	B	$\theta^1_{B\|A}$
T	T	.21661
T	F	.78339
F	T	.70741
F	F	.29259

A	C	$\theta^1_{C\|A}$
T	T	.18410
T	F	.81590
F	T	.80713
F	F	.19287

Every parent instantiation has $Pr_{\mathcal{D},\theta^0}(\mathbf{u}) > 0$, so by Theorem 17.5 these are the unique maximizers of $ELL(\theta|\mathcal{D}, \theta^0)$.

Problem 17.7. Consider Exercise 17.5, that is, the Bayesian network with edges $A \to B$ and $A \to C$ whose CPTs are $\theta_a = .3$; $\theta_{b|a} = .5$, $\theta_{b|\bar{a}} = .8$; $\theta_{c|a} = .1$, $\theta_{c|\bar{a}} = .5$, together with the incomplete data set

Case	A	B	C
1	F	?	T
2	F	T	T
3	?	F	T
4	?	T	F
5	T	F	?

and assume that the given CPTs are those used by the gradient ascent approach for estimating parameters. Decide whether this approach increases or decreases the values of parameters $\theta_{\bar{a}}$, $\theta_{b|a}$, and $\theta_{\bar{c}|a}$ in its first iteration. Use the soft-max search space to impose parameter constraints.

Solution. All three decrease, each of the three parameters exceeding the EM estimate for its row. Under the soft-max reparameterization $\theta_{x|\mathbf{u}} = e^{\tau_{x|\mathbf{u}}} / \sum_{x^\star} e^{\tau_{x^\star|\mathbf{u}}}$ of Section 17.3.2 gradient ascent steps

$\tau_{x|\mathbf{u}}$ by $\eta\,\partial LL/\partial\tau_{x|\mathbf{u}}$, so the sign of that gradient is the answer. The soft-max derivatives are

$$\frac{\partial\theta_{x|\mathbf{u}}}{\partial\tau_{x|\mathbf{u}}} = \theta_{x|\mathbf{u}}(1-\theta_{x|\mathbf{u}}), \qquad \frac{\partial\theta_{x^\star|\mathbf{u}}}{\partial\tau_{x|\mathbf{u}}} = -\,\theta_{x|\mathbf{u}}\,\theta_{x^\star|\mathbf{u}} \quad (x^\star \neq x),$$

by Exercise 17.10 and its off-diagonal companion, $\tau_{x|\mathbf{u}}$ leaving every other CPT row untouched; and $\partial LL/\partial\theta_{x|\mathbf{u}} = N\,Pr_{\mathcal{D},\theta}(x\mathbf{u})/\theta_{x|\mathbf{u}}$ by Theorem 17.8 (Equation 17.10) with Theorem 17.4. Chaining through the whole row, the θ factors cancel:

$$\begin{aligned}\frac{\partial LL}{\partial\tau_{x|\mathbf{u}}} &= \sum_{x^\star}\frac{\partial LL}{\partial\theta_{x^\star|\mathbf{u}}}\frac{\partial\theta_{x^\star|\mathbf{u}}}{\partial\tau_{x|\mathbf{u}}}\\ &= N\,Pr_{\mathcal{D},\theta}(x\mathbf{u}) - N\theta_{x|\mathbf{u}}\sum_{x^\star}Pr_{\mathcal{D},\theta}(x^\star\mathbf{u})\\ &= N\,Pr_{\mathcal{D},\theta}(\mathbf{u})\big[\,Pr_{\mathcal{D},\theta}(x|\mathbf{u}) - \theta_{x|\mathbf{u}}\,\big].\end{aligned}$$

Every parent instantiation here has $Pr_{\mathcal{D},\theta}(\mathbf{u}) > 0$, so $\theta_{x|\mathbf{u}}$ increases exactly when the EM estimate $Pr_{\mathcal{D},\theta}(x|\mathbf{u})$ of Definition 17.3 exceeds it, the two gradients of a binary row being negatives of each other. (Reading Equation 17.8 as the single product $(\partial LL/\partial\theta_{x|\mathbf{u}})(\partial\theta_{x|\mathbf{u}}/\partial\tau_{x|\mathbf{u}})$ instead gives growth iff $Pr_{\mathcal{D},\theta}(x\mathbf{u})(1-\theta_{x|\mathbf{u}}) > Pr_{\mathcal{D},\theta}(\bar{x}\mathbf{u})(1-\theta_{\bar{x}|\mathbf{u}})$, the same criterion.) Exercises 17.5 and 17.6 supply the conditionals:

$$\begin{aligned}Pr_{\mathcal{D},\theta}(\bar{a}) &= 4936/7055 \approx .69965,\\ Pr_{\mathcal{D},\theta}(b|a) &= 459/2119 \approx .21661,\\ Pr_{\mathcal{D},\theta}(\bar{c}|a) &= 17289/21190 \approx .81590.\end{aligned}$$

Each falls below the current value, so each gradient is negative:

parameter	current θ	$Pr_{\mathcal{D},\theta}(x \mid \mathbf{u})$	difference	effect
$\theta_{\bar{a}}$	.7	.69965	−.00035	decrease
$\theta_{b\|a}$	.5	.21661	−.28339	decrease
$\theta_{\bar{c}\|a}$	.9	.81590	−.08410	decrease

Explicitly, with $N = 5$, $Pr_{\mathcal{D},\theta}(\top) = 1$ for the root and $Pr_{\mathcal{D},\theta}(a) \approx .30035$ for the rows conditioned on a:

$$\begin{aligned}\frac{\partial LL}{\partial\tau_{\bar{a}}} &= 5(1)(.699646 - .7) = -.00177,\\ \frac{\partial LL}{\partial\tau_{b|a}} &= 5(.30035)(.21661 - .5) = -.42558,\\ \frac{\partial LL}{\partial\tau_{\bar{c}|a}} &= 5(.30035)(.81590 - .9) = -.12629.\end{aligned}$$

16.2 Exercises 17.8–17.14

Problem 17.8. Consider a complete data set D and let G_1 and G_2 be two complete DAGs (that is, DAGs over the variables of D in which every pair of nodes is connected by an edge). Prove or disprove

$$\mathrm{LL}(G_1 \mid D) = \mathrm{LL}(G_2 \mid D).$$

If the equality does not hold, provide a counterexample.

Solution. The equality always holds: every complete DAG has the same log-likelihood given a complete data set, namely $-N \cdot \mathrm{ENT}_D(\mathbf{X})$.
A complete DAG G over $\mathbf{X} = X_1, \ldots, X_n$ has a unique topological order $X_{\sigma(1)}, \ldots, X_{\sigma(n)}$ (adjacency makes any two nodes comparable), with parent sets $\mathbf{U}_i = \{X_{\sigma(1)}, \ldots, X_{\sigma(i-1)}\}$, and ML estimates

$\theta^{ml}_{x|u} = \Pr_D(x \mid u)$ by Theorem 17.1. Hence for any full instantiation $\mathbf{x}$,

$$\begin{aligned}\Pr_{\theta^{ml}}(\mathbf{x}) &= \prod_{i=1}^{n} \theta^{ml}_{x_{\sigma(i)}|\mathbf{u}_i} \\ &= \prod_{i=1}^{n} \Pr_D\big(x_{\sigma(i)} \mid \mathbf{u}_i\big) \\ &= \Pr_D(\mathbf{x}),\end{aligned}$$

the last step being the chain rule for $\Pr_D$ in the order σ (legitimate since $\Pr_D(\mathbf{x}) > 0$ forces $\Pr_D(\mathbf{u}_i) > 0$, while rows conditioned on zero-count instantiations do not affect the likelihood by Exercise 17.3). So G with its ML parameters induces $\Pr_D$ whatever σ is, and

$$\begin{aligned}\mathrm{LL}(G \mid D) &= \sum_{i=1}^{N} \log \Pr_{\theta^{ml}}(d_i) = \sum_{i=1}^{N} \log \Pr_D(d_i) \\ &= N \sum_{\mathbf{x}} \Pr_D(\mathbf{x}) \log \Pr_D(\mathbf{x}) \\ &= -N \cdot \mathrm{ENT}_D(\mathbf{X}),\end{aligned}$$

grouping the N cases by the instantiation they realize via $D\#(\mathbf{x}) = N \Pr_D(\mathbf{x})$. The right-hand side mentions only D, so $\mathrm{LL}(G_1 \mid D) = \mathrm{LL}(G_2 \mid D)$.
Method (2): by Theorem 17.3,

$$\mathrm{LL}(G \mid D) = -N \sum_{i=1}^{n} \mathrm{ENT}_D(X_{\sigma(i)} \mid \mathbf{U}_i).$$

With $\mathrm{ENT}(X \mid \mathbf{U}) = \mathrm{ENT}(X\mathbf{U}) - \mathrm{ENT}(\mathbf{U})$ and $\mathbf{U}_{i+1} = \mathbf{U}_i \cup \{X_{\sigma(i)}\}$ the sum telescopes:

$$\begin{aligned}\sum_{i=1}^{n} \mathrm{ENT}_D(X_{\sigma(i)} \mid \mathbf{U}_i) &= \sum_{i=1}^{n} [\mathrm{ENT}_D(\mathbf{U}_{i+1}) - \mathrm{ENT}_D(\mathbf{U}_i)] \\ &= \mathrm{ENT}_D(X_1, \ldots, X_n),\end{aligned}$$

since $\mathbf{U}_1 = \emptyset$ has zero entropy and $\mathbf{U}_{n+1} = \mathbf{X}$; the value is invariant under σ.

Problem 17.9. What happens if we apply the EM algorithm to a complete data set? That is, what estimates does it return as a function of the initial estimates with which it starts?

Solution. On complete data EM ignores its initial estimates entirely and returns the ML estimates $\theta^{ml}_{x|u} = \Pr_D(x \mid u)$ after a single iteration, after which it sits at a fixed point.
Let θ^0 be any start with $\Pr_{\theta^0}(d_i) > 0$ for every case (needed for Definition 17.3 to apply; hence the usual strictly positive initialization). Each complete case instantiates every variable, so the missing set $\mathbf{C}_i$ of Definition 17.2 is empty, d_i is its own sole completion, and $\Pr_{\theta^0}(xu \mid d_i) \in \{0, 1\}$ according as $d_i \models xu$ or not. Hence the E-step returns the ordinary empirical distribution,

$$\begin{aligned}\Pr_{D,\theta^0}(\alpha) &= \frac{1}{N} \sum_{i=1}^{N} \Pr_{\theta^0}(\alpha \mid d_i) \\ &= \frac{D\#(\alpha)}{N} = \Pr_D(\alpha),\end{aligned}$$

by Theorem 17.4, with θ^0 dropped out. The M-step (17.7) then gives

$$\theta^1_{x|u} = \frac{\sum_i \Pr_{\theta^0}(xu \mid d_i)}{\sum_i \Pr_{\theta^0}(u \mid d_i)} = \frac{D\#(xu)}{D\#(u)} = \Pr_D(x \mid u),$$

the ML estimate of Theorem 17.1. Nothing beyond positivity of θ^0 was used, and $\theta^1 = \theta^{ml}$ is itself positive on every case (each d_i has $D\#(d_i) \geq 1$), so repeating gives $\theta^2 = \theta^1$: EM converges in one iteration, onto the unique maximum of the log-likelihood, as Theorem 17.7 predicts. (Where $D\#(u) = 0$ both parts of (17.7) vanish and $\theta^1_{x|u}$ stays at $\theta^0_{x|u}$, harmlessly: by Exercise 17.3 the likelihood ignores those parameters.)

Problem 17.10. When performing gradient ascent on the log-likelihood function one must respect the constraints $\theta_{x|u} \in [0,1]$ and $\sum_x \theta_{x|u} = 1$. Section 17.3.2 handles this by searching in the **softmax space**: for each parameter $\theta_{x|u}$ one introduces a variable $\tau_{x|u}$ with values in $(-\infty, \infty)$ and defines

$$\theta_{x|u} \stackrel{\text{def}}{=} \frac{e^{\tau_{x|u}}}{\sum_{x^\star} e^{\tau_{x^\star|u}}}.$$

Gradient ascent then uses the gradient of Equation 17.8,

$$\frac{\partial \mathrm{LL}(\theta \mid D)}{\partial \tau_{x|u}} = \frac{\partial \mathrm{LL}(\theta \mid D)}{\partial \theta_{x|u}} \cdot \frac{\partial \theta_{x|u}}{\partial \tau_{x|u}}.$$

Compute the derivative $\dfrac{\partial \theta_{x|u}}{\partial \tau_{x|u}}$ given in Equation 17.8.

Solution. $\partial\theta_{x|u}/\partial\tau_{x|u} = \theta_{x|u}(1-\theta_{x|u})$. Writing $Z_u = \sum_{x^\star} e^{\tau_{x^\star|u}}$, so that $\partial Z_u/\partial\tau_{x|u} = e^{\tau_{x|u}}$, the quotient rule gives

$$\begin{aligned}\frac{\partial \theta_{x|u}}{\partial \tau_{x|u}} &= \frac{e^{\tau_{x|u}} Z_u - e^{2\tau_{x|u}}}{Z_u^2} \\ &= \frac{e^{\tau_{x|u}}}{Z_u}\left(1 - \frac{e^{\tau_{x|u}}}{Z_u}\right) \\ &= \theta_{x|u}\big(1 - \theta_{x|u}\big).\end{aligned}$$

Substituting this and Theorem 17.8 into (17.8),

$$\frac{\partial \mathrm{LL}(\theta \mid D)}{\partial \tau_{x|u}} = \theta_{x|u}(1-\theta_{x|u}) \sum_{i=1}^{N} \frac{\Pr_\theta(xu \mid d_i)}{\theta_{x|u}} = (1-\theta_{x|u}) \sum_{i=1}^{N} \Pr_\theta(xu \mid d_i).$$

The map is not coordinate-wise: $\tau_{x|u}$ moves every $\theta_{x^\star|u}$ of the same column, with

$$\frac{\partial \theta_{y|u^\star}}{\partial \tau_{x|u}} = \delta_{uu^\star}\, \theta_{y|u}\big(\delta_{xy} - \theta_{x|u}\big)$$

(the $\delta_{uu^\star}$ because a different parent instantiation uses disjoint τ variables), so the exact chain rule is the sum $\sum_{x^\star}(\partial \mathrm{LL}/\partial\theta_{x^\star|u})(\partial\theta_{x^\star|u}/\partial\tau_{x|u})$, which by Theorem 17.8 collapses to $\sum_i [\Pr_\theta(xu \mid d_i) - \theta_{x|u} \Pr_\theta(u \mid d_i)]$ (Exercise 17.7), rather than the single product displayed in (17.8).

Problem 17.11. Compute an ML tree structure for the data set in Exercise 17.1. What is the total number of ML tree structures for this data set? The data set of Exercise 17.1 is over the binary variables A, B, C, D:

Case	A	B	C	D
1	T	F	F	F
2	T	F	F	T
3	F	F	T	F
4	T	T	F	T
5	F	F	T	T
6	F	T	T	F
7	F	T	T	T
8	T	F	F	T
9	F	F	T	F
10	T	T	T	T

Recall that a tree structure is a DAG in which every node has at most one parent and whose skeleton is a spanning tree; such a structure is obtained by choosing a root and directing all edges away from it.

Solution. One ML tree is $A \to C$, $A \to D$, $C \to B$, and there are 8 in all. By Theorem 17.10 the ML trees are exactly the orientations of the maximum spanning trees of the complete graph on A, B, C, D weighted by MI_D, since $\mathrm{tScore}(G \mid D) = \sum_{U \to X} \mathrm{MI}_D(X, U)$. From the $N = 10$ cases,

$$D\#(A{=}T) = 5, \quad D\#(B{=}T) = 4,$$
$$D\#(C{=}T) = 6, \quad D\#(D{=}T) = 6,$$

and the six pairwise count tables (rows are the first variable's value):

pair	TT	TF	FT	FF
A,B	2	3	2	3
A,C	1	4	5	0
A,D	4	1	2	3
B,C	3	1	3	3
B,D	3	1	3	3
C,D	3	3	3	1

With $\mathrm{MI}_D(X,U) = \sum_{x,u} \Pr_D(xu) \log_2 \frac{\Pr_D(xu)}{\Pr_D(x)\Pr_D(u)}$: the pair A, B factors exactly ($0.2 = 0.5 \times 0.4$, and likewise the other three entries), so $\mathrm{MI}_D(A,B) = 0$. For A, C:

$$\begin{aligned}\mathrm{MI}_D(A,C) &= 0.1 \log_2 \tfrac{0.1}{0.3} + 0.4 \log_2 \tfrac{0.4}{0.2} + 0.5 \log_2 \tfrac{0.5}{0.3} \\ &= 0.4 + \tfrac{1}{2}\log_2 5 - \tfrac{3}{5}\log_2 3 \;=\; 0.60999.\end{aligned}$$

For A, D:

$$\begin{aligned}\mathrm{MI}_D(A,D) &= 0.4 \log_2 \tfrac{4}{3} + 0.1 \log_2 \tfrac{1}{2} + 0.2 \log_2 \tfrac{2}{3} + 0.3 \log_2 \tfrac{3}{2} \\ &= 0.12451.\end{aligned}$$

The pairs B, C, B, D, C, D share the count multiset $\{3, 3, 3, 1\}$ against marginals $\{6, 4\}$ and $\{4, 6\}$, hence share a value:

$$\begin{aligned}\mathrm{MI}_D(B,C) &= \mathrm{MI}_D(B,D) = \mathrm{MI}_D(C,D) \\ &= 2\big(0.3 \log_2 \tfrac{0.3}{0.24}\big) + 0.3 \log_2 \tfrac{0.3}{0.36} \\ &\quad + 0.1 \log_2 \tfrac{0.1}{0.16} \\ &= 0.04644.\end{aligned}$$

Collected:

edge	MI_D
A-B	0.00000
A-C	0.60999
A-D	0.12451
B-C	0.04644
B-D	0.04644
C-D	0.04644

Kruskal on these weights takes A-C, then A-D; the remaining edge must attach B, and of the candidates B-C, B-D (both 0.04644) and A-B (0) the first two tie, while C-D, tied in weight, would close a cycle. So the maximum spanning trees are

$$T_1 = \{A\text{-}C,\ A\text{-}D,\ B\text{-}C\} \quad \text{or} \quad T_2 = \{A\text{-}C,\ A\text{-}D,\ B\text{-}D\},$$

both of cost

$$0.60999 + 0.12451 + 0.04644 = 0.78094.$$

Enumerating the $4^{4-2} = 16$ spanning trees on four labeled nodes confirms T_1, T_2 are the only maximizers (next best 0.73450, the star at A). Rooting T_1 at A gives $A \to C$, $A \to D$, $C \to B$, whose ML parameters $\theta^{ml}_{x|u} = \Pr_D(x \mid u)$ (Theorem 17.1) are read off the counts:

A	θ_a
T	1/2
F	1/2

A	C	$\theta_{c\|a}$
T	T	1/5
T	F	4/5
F	T	1
F	F	0

A	D	$\theta_{d\|a}$
T	T	4/5
T	F	1/5
F	T	2/5
F	F	3/5

C	B	$\theta_{b\|c}$
T	T	1/2
T	F	1/2
F	T	1/4
F	F	3/4

Each of the two maximum spanning trees yields exactly $n = 4$ directed trees, one per root, pairwise distinct because different roots orient differently, and the two skeletons differ; hence $2 \times 4 = 8$ ML tree structures. For T_1:

$$\begin{aligned}
\text{root } A: &\quad A \to C,\ A \to D,\ C \to B \\
\text{root } B: &\quad B \to C,\ C \to A,\ A \to D \\
\text{root } C: &\quad C \to A,\ C \to B,\ A \to D \\
\text{root } D: &\quad D \to A,\ A \to C,\ C \to B
\end{aligned}$$

and the four for T_2 are obtained by replacing the edge between B and C with one between B and D.

Problem 17.12. Consider a Bayesian network structure consisting of the single edge $X \to Y$ (so the parameters are θ_x for the root X and $\theta_{y|x}$ for Y), and a data set D of size N in which X is hidden and Y is observed in every case of D. Show that ML estimates are characterized by

$$\sum_x \theta_x \, \theta_{y|x} \ = \ \frac{D\#(y)}{N}$$

for all values y; that is, ML estimates are those that ensure $\Pr(y) = D\#(y)/N$ for all y.

Solution. The log-likelihood depends on θ only through the induced marginal $\Pr_\theta(Y)$, which Lemma

17.1 maximizes uniquely at $\Pr_D(Y)$. Every case d_i is an instantiation y_i of Y, so

$$\begin{aligned}\mathrm{LL}(\theta \mid D) &= \sum_{i=1}^{N} \log \Pr_\theta(d_i) = \sum_{i=1}^{N} \log \Pr_\theta(y_i) \\ &= \sum_y D\#(y) \log \Pr_\theta(y) = N \sum_y \Pr_D(y) \log \Pr_\theta(y),\end{aligned}$$

grouping the cases by the value of Y they report, with $\Pr_D(y) = D\#(y)/N$; and summing out the hidden root, $\Pr_\theta(y) = \sum_x \theta_x \theta_{y|x}$.
Lemma 17.1, applied to the fixed distribution $\Pr_D(Y)$ against the variable $\Pr_\theta(Y)$, bounds $\sum_y \Pr_D(y) \log \Pr_\theta(y) \le \sum_y \Pr_D(y) \log \Pr_D(y)$ with equality iff $\Pr_\theta(Y) = \Pr_D(Y)$, so

$$\mathrm{LL}(\theta \mid D) \;\le\; -N \cdot \mathrm{ENT}_D(Y),$$

with equality exactly when $\sum_x \theta_x \theta_{y|x} = D\#(y)/N$ for all y. The bound is attained, since $\theta_{y|x} = D\#(y)/N$ for every x is a legitimate CPT meeting it whatever the root parameters, so $-N\,\mathrm{ENT}_D(Y)$ is the maximum and the displayed condition characterizes the ML estimates.

Problem 17.13. Consider a Bayesian network structure with edges $X \to Y$ and $Y \to Z$ (a chain, with parameters θ_x, $\theta_{y|x}$ and $\theta_{z|y}$), and a data set D of size N in which Y is hidden yet X and Z are observed in every case of D. Provide a characterization of the ML estimates for this problem, together with some concrete parameter estimates that satisfy the characterization.

Solution. Assuming $|Y| \ge \min(|X|, |Z|)$ (so in particular for binary variables), θ is an ML estimate iff it reproduces the empirical joint over the observed variables:

$$\Pr_\theta(x, z) \;=\; \theta_x \sum_y \theta_{y|x} \theta_{z|y} \;=\; \frac{D\#(xz)}{N} \quad \text{for all } x, z,$$

equivalently, if and only if

$$\theta_x = \frac{D\#(x)}{N} \quad \text{and} \quad \sum_y \theta_{y|x} \theta_{z|y} = \frac{D\#(xz)}{D\#(x)}$$

for all x with $D\#(x) > 0$ and all z (the rows $\theta_{Y|x}$ at $D\#(x) = 0$ being unconstrained); the maximum log-likelihood is $-N\,\mathrm{ENT}_D(X, Z)$.
Each case d_i is an instantiation $x_i z_i$ of X and Z only, so, writing $M(z \mid x) \stackrel{\text{def}}{=} \sum_y \theta_{y|x} \theta_{z|y}$ — a distribution over Z for each x, with $\Pr_\theta(x, z) = \theta_x M(z \mid x)$ — and splitting the logarithm,

$$\mathrm{LL}(\theta \mid D) = \sum_{i=1}^{N} \log \Pr_\theta(x_i z_i) = \sum_{x,z} D\#(xz) \log \Pr_\theta(x, z),$$

$$\mathrm{LL}(\theta \mid D) = \underbrace{\sum_x D\#(x) \log \theta_x}_{S_1} \;+\; \underbrace{\sum_x \sum_z D\#(xz) \log M(z \mid x)}_{S_2}.$$

The two groups are disjoint — S_1 involves only θ_x, S_2 only $\theta_{y|x}, \theta_{z|y}$ — so they maximize independently. By Lemma 17.1 with the fixed $\Pr_D(X)$ against the variable θ_X,

$$S_1 = N \sum_x \Pr_D(x) \log \theta_x \le N \sum_x \Pr_D(x) \log \Pr_D(x),$$

with equality only for $\theta_x = D\#(x)/N$. Grouping S_2 by x, with $\Pr_D(z \mid x) = D\#(xz)/D\#(x)$,

$$S_2 = \sum_{x\,:\,D\#(x)>0} D\#(x) \sum_z \Pr_D(z \mid x) \log M(z \mid x).$$

Each inner sum is again of Lemma 17.1's form, so $S_2 \le \sum_x D\#(x) \sum_z \Pr_D(z \mid x) \log \Pr_D(z \mid x)$ with equality iff $M(z \mid x) = \Pr_D(z \mid x)$ for every x with $D\#(x) > 0$. That M is realizable by the chain: the realizable M are exactly the products

$$M = AB, \qquad A = [\theta_{y|x}]_{|X| \times |Y|}, \qquad B = [\theta_{z|y}]_{|Y| \times |Z|},$$

with A and B row-stochastic. Two cases realize $[\Pr_D(z \mid x)]$:

- (i) $|Y| \ge |X|$: fix an injection $x_j \mapsto y_j$, set $\theta_{y_j|x_j} = 1$ (all other entries of A zero, so A selects rows of B), and set row y_j of B to $\Pr_D(\cdot \mid x_j)$, the leftover rows arbitrary.
- (ii) $|Y| \ge |Z|$: fix an injection $z_k \mapsto y_k$, set $\theta_{z_k|y_k} = 1$ (so B embeds Z into Y), and set $A = [\Pr_D(z \mid x)]$ padded with zero columns.

Either way $AB = [\Pr_D(z \mid x)]$ and the bound is attained. Combining the two bounds,

$$\mathrm{LL}^{\max} = \sum_{x,z} D\#(xz) \log \frac{D\#(xz)}{N} = -N \,\mathrm{ENT}_D(X, Z),$$

attained exactly when $\theta_x = D\#(x)/N$ and $M(z \mid x) = \Pr_D(z \mid x)$, i.e. when $\Pr_\theta(x, z) = D\#(xz)/N$. For concrete estimates take $|Y| \ge |X|$ and let Y be a deterministic copy of X:

$$\theta_{y_j|x_j} = 1, \qquad \theta_{z|y_j} = \frac{D\#(x_j z)}{D\#(x_j)}, \qquad \theta_{x_j} = \frac{D\#(x_j)}{N}.$$

Then

$$\Pr_\theta(x_j, z) = \frac{D\#(x_j)}{N} \cdot 1 \cdot \frac{D\#(x_j z)}{D\#(x_j)} = \frac{D\#(x_j z)}{N},$$

as required. For a numeric instance, let all three be binary with $N = 10$ and counts

X	Z	count
T	T	4
T	F	1
F	T	2
F	F	3

Then the estimates are

X	θ_x
T	.5
F	.5

X	Y	$\theta_{y\|x}$
T	T	1
T	F	0
F	T	0
F	F	1

Y	Z	$\theta_{z\|y}$
T	T	.8
T	F	.2
F	T	.4
F	F	.6

matching the empirical joint (Check!).

Problem 17.14. Consider the naive Bayes network structure of Figure 17.15: a single root node C with m children $A_1, A_2, \ldots, A_m$, so the edges are $C \to A_i$ for $i = 1, \ldots, m$ and there are no other edges. Suppose that D is a data set of size N in which variable C is hidden and variables $A_1, \ldots, A_m$ are observed in all cases. Show that if EM is applied to this problem with parameters having uniform

initial values, then it will converge in one step, returning the following estimates:

$$\theta_c = \frac{1}{|C|}, \qquad \theta_{a_i|c} = \frac{D\#(a_i)}{N}.$$

Here $|C|$ is the cardinality of variable C, N is the size of the given data set, and $D\#(a_i)$ is the number of cases in the data set that contain instantiation a_i of variable A_i.

Solution. Uniform parameters make the posterior over the hidden class uniform in every case, which collapses both steps. With $\theta^0_c = 1/|C|$, $\theta^0_{a_i|c} = 1/|A_i|$ and a case $d_j = a^j_1, \ldots, a^j_m$ instantiating all attributes, the chain rule gives

$$\Pr_{\theta^0}(c, d_j) = \theta^0_c \prod_{i=1}^{m} \theta^0_{a^j_i|c} = \frac{1}{|C|} \prod_{i=1}^{m} \frac{1}{|A_i|},$$

independent of c; summing over c gives $\Pr_{\theta^0}(d_j) = \prod_i 1/|A_i| > 0$, so conditioning is legitimate and $\Pr_{\theta^0}(c \mid d_j) = 1/|C|$ throughout. In the M-step (17.7) the root has empty parent set, so the denominator is N:

$$\theta^1_c = \frac{\sum_{j=1}^{N} \Pr_{\theta^0}(c \mid d_j)}{N} = \frac{N \cdot \frac{1}{|C|}}{N} = \frac{1}{|C|}.$$

Since A_i is observed in every case, d_j already determines it, so

$$\Pr_{\theta^0}(a_i c \mid d_j) = \begin{cases} \Pr_{\theta^0}(c \mid d_j) = \frac{1}{|C|}, & \text{if } d_j \models a_i, \\ 0, & \text{otherwise,} \end{cases}$$

Hence

$$\theta^1_{a_i|c} = \frac{\sum_j \Pr_{\theta^0}(a_i c \mid d_j)}{\sum_j \Pr_{\theta^0}(c \mid d_j)} = \frac{\sum_{j\,:\,d_j \models a_i} \frac{1}{|C|}}{\sum_{j=1}^{N} \frac{1}{|C|}}$$
$$= \frac{\frac{1}{|C|} D\#(a_i)}{\frac{1}{|C|} N} = \frac{D\#(a_i)}{N}.$$

as claimed. Repeating the E-step at θ^1,

$$\Pr_{\theta^1}(c, d_j) = \frac{1}{|C|} \prod_{i=1}^{m} \frac{D\#(a^j_i)}{N},$$

again independent of c, and positive because d_j contains a^j_i, so $D\#(a^j_i) \geq 1$. Thus $\Pr_{\theta^1}(c \mid d_j) = 1/|C|$ once more, and since the M-step used the current parameters only through this uniform posterior, $\theta^2 = \theta^1$.

16.3 Exercises 17.15–17.20

Problem 17.15. Consider the naive Bayes network structure in Figure 17.15 and let $\Pr_\theta(\cdot)$ be the distribution induced by this structure and parametrization θ.
Figure 17.15 is the naive Bayes structure over variables $C, A_1, \ldots, A_m$: node C is a root with no parents, and the only edges are $C \to A_1$, $C \to A_2$, …, $C \to A_m$. Thus the network has one CPT Θ_C with parameters θ_c and m CPTs $\Theta_{A_i|C}$ with parameters $\theta_{a_i|c}$.
Let us refer to variable C as the class variable, variables $A_1, \ldots, A_m$ as the attributes, and each instantiation $a_1, \ldots, a_m$ as an instance. Suppose that we now use this network as a classifier where we assign to each instance $a_1, \ldots, a_m$ the class c that maximizes $\Pr_\theta(c \mid a_1, \ldots, a_m)$. Show that when

variable C is binary, the class of instance $a_1, \ldots, a_m$ is c if

$$\log \theta_c + \sum_{i=1}^{m} \log \theta_{a_i|c} \;>\; \log \theta_{\bar{c}} + \sum_{i=1}^{m} \log \theta_{a_i|\bar{c}}.$$

Note: This is known as a naive Bayes classifier.

Solution. The families of Figure 17.15 are C and A_iC, so for the instance $\alpha = a_1, \ldots, a_m$ the chain rule gives

$$\Pr_\theta(c, \alpha) = \theta_c \prod_{i=1}^{m} \theta_{a_i|c}, \qquad \Pr_\theta(\bar{c}, \alpha) = \theta_{\bar{c}} \prod_{i=1}^{m} \theta_{a_i|\bar{c}}.$$

Since C is binary, α is classified c exactly when $\Pr_\theta(c \mid \alpha) > \Pr_\theta(\bar{c} \mid \alpha)$; multiplying through by the common normalizing constant $\Pr_\theta(\alpha)$ (positive, else α has no posterior) compares the two joints instead, and the strictly increasing log gives

$$\log \theta_c + \sum_{i=1}^{m} \log \theta_{a_i|c} \;>\; \log \theta_{\bar{c}} + \sum_{i=1}^{m} \log \theta_{a_i|\bar{c}},$$

the stated rule. (With $\log 0 = -\infty$ the equivalence survives zero parameters on one side.)

Problem 17.16. Consider Exercise 17.15 (the naive Bayes structure of Figure 17.15, with class variable C as the single root and attributes $A_1, \ldots, A_m$ as its children). Let D be a complete data set that contains the cases $c^i, a_1^i, \ldots, a_m^i$ for $i = 1, \ldots, N$, and let $\Pr_D(\cdot)$ be the empirical distribution induced by data set D. Define the conditional log-likelihood function as

$$CLL(D \mid \theta) \;\stackrel{\text{def}}{=}\; \sum_{i=1}^{N} \log \Pr_\theta(c^i \mid a_1^i, \ldots, a_m^i).$$

Define also the conditional KL divergence as

$$\sum_{a_1,\ldots,a_m} \Pr_D(a_1, \ldots, a_m) \cdot KL\big(\Pr_D(C \mid a_1, \ldots, a_m),\, \Pr_\theta(C \mid a_1, \ldots, a_m)\big).$$

Show that a parametrization θ maximizes the conditional log-likelihood function iff it minimizes the conditional KL divergence.

Solution. The two criteria differ by a constant: $CLL(D \mid \theta) = -N(ENT_D(C \mid \mathbf{A}) + CKL(\theta))$, whose first term is free of θ.
Write α for an instance and $\mathbf{A}$ for $A_1, \ldots, A_m$; sums range over α with $\Pr_D(\alpha) > 0$, where we may assume $\Pr_\theta(\alpha) > 0$ (else $CLL = -\infty$ and $CKL = +\infty$, and both criteria reject θ). Grouping the cases by the instantiation they realize, each contributing $D\#(c, \alpha) = N \Pr_D(c, \alpha)$ identical terms,

$$\begin{aligned} CLL(D \mid \theta) &= \sum_{i=1}^{N} \log \Pr_\theta(c^i \mid \alpha^i) \\ &= N \sum_{c,\alpha} \Pr_D(c, \alpha) \log \Pr_\theta(c \mid \alpha) \\ &= N \sum_{\alpha} \Pr_D(\alpha) \sum_{c} \Pr_D(c \mid \alpha) \log \Pr_\theta(c \mid \alpha), \end{aligned}$$

the last line by the chain rule for $\Pr_D$. Expanding the KL divergence by its definition (Appendix B),

splitting the logarithm of the ratio and weighting by $\Pr_D(\alpha)$,

$$\begin{aligned} CKL(\theta) &\stackrel{\text{def}}{=} \sum_{\alpha} \Pr_D(\alpha)\, KL\big(\Pr_D(C \mid \alpha), \Pr_\theta(C \mid \alpha)\big) \\ &= \sum_{\alpha} \Pr_D(\alpha) \sum_{c} \Pr_D(c \mid \alpha) \log \Pr_D(c \mid \alpha) \\ &\quad - \sum_{\alpha} \Pr_D(\alpha) \sum_{c} \Pr_D(c \mid \alpha) \log \Pr_\theta(c \mid \alpha). \end{aligned}$$

The first double sum is $-ENT_D(C \mid \mathbf{A})$ by definition; the second is $CLL(D \mid \theta)/N$ by the display above. Hence

$$CLL(D \mid \theta) \;=\; -N\big(ENT_D(C \mid \mathbf{A}) + CKL(\theta)\big),$$

affine in $CKL(\theta)$ with negative slope $-N$ and θ-free intercept. So $CLL(D \mid \theta) \geq CLL(D \mid \theta')$ iff $CKL(\theta) \leq CKL(\theta')$, and the maximizers of the one are the minimizers of the other.

Problem 17.17. Consider Exercise 17.16: the naive Bayes structure of Figure 17.15 with class variable C as the single root and attributes $A_1, \ldots, A_m$ as its children, a complete data set D containing the cases $c^i, a_1^i, \ldots, a_m^i$ for $i = 1, \ldots, N$, and the conditional log-likelihood

$$CLL(D \mid \theta) = \sum_{i=1}^{N} \log \Pr_\theta(c^i \mid a_1^i, \ldots, a_m^i).$$

Let D' be an incomplete data set that is obtained from data set D by removing the values of class variable C; that is, D' contains the cases $a_1^i, \ldots, a_m^i$ for $i = 1, \ldots, N$. Show that

$$LL(D \mid \theta) \;=\; CLL(D \mid \theta) + LL(D' \mid \theta).$$

Solution. The identity is the chain rule $\Pr_\theta(c^i, \alpha^i) = \Pr_\theta(c^i \mid \alpha^i) \Pr_\theta(\alpha^i)$, taken in logarithms and summed over cases.
Write α^i for the instance $a_1^i, \ldots, a_m^i$; case i of D is c^i, α^i, while case i of D' records only α^i, whose probability is the marginal $\Pr_\theta(\alpha^i) = \sum_c \Pr_\theta(c, \alpha^i)$ (Section 17.3). Assuming $\Pr_\theta(\alpha^i) > 0$ for every case,

$$\begin{aligned} LL(D \mid \theta) &= \sum_{i=1}^{N} \log \Pr_\theta(c^i, \alpha^i) \\ &= \sum_{i=1}^{N} \log \Pr_\theta(c^i \mid \alpha^i) \;+\; \sum_{i=1}^{N} \log \Pr_\theta(\alpha^i) \\ &= CLL(D \mid \theta) + LL(D' \mid \theta), \end{aligned}$$

as required. (If $\Pr_\theta(\alpha^i) = 0$ for some case then $\Pr_\theta(c^i, \alpha^i) = 0$ too, and the identity reads $-\infty = -\infty + (-\infty)$ under the convention $CLL(D \mid \theta) = -\infty$.)

Problem 17.18. Consider the structure in Figure 17.6(b) and a data set with a single case:

$$C = \text{no}, \qquad T = ?, \qquad I = \text{yes}.$$

Figure 17.6(b) is the network structure that explicates a *nonignorable* missing-data mechanism for the medical example of Section 17.3.3. It has three nodes and two edges:

$$C \longrightarrow T \longrightarrow I.$$

Here C is the medical condition with values yes and no, T is the test for that condition with values $+ve$ and $-ve$, and I is the missing-data indicator for T (Definition 17.4), with $I =$ yes meaning that

the value of T is missing from the data set. The parameter sets are θ_C, $\theta_{T|C}$ and $\theta_{I|T}$; in particular the missingness of the test result depends on the test result itself, which is why the mechanism is not MAR (Definition 17.5) and cannot be ignored.
Show that the ML estimates are characterized as follows:

- $\theta_{C=\text{no}}$ is 1.
- One of the following must be true for the parameters of test T and missing-data indicator I:
  - $\theta_{T=-ve|C=\text{no}} = 1$ and $\theta_{I=\text{yes}|T=-ve} = 1$
  - $\theta_{T=+ve|C=\text{no}} = 1$ and $\theta_{I=\text{yes}|T=+ve} = 1$
  - $\theta_{I=\text{yes}|T=-ve} = \theta_{I=\text{yes}|T=+ve} = 1$.

Solution. The likelihood is $L(\theta \mid D) = \Pr_\theta(C = \text{no}, I = \text{yes})$, the single case recording $C = \text{no}$ together with $I = \text{yes}$ (that T is missing), and its maximum value is 1. Summing the unobserved T out of the structure $C \to T \to I$,

$$\begin{aligned} L(\theta \mid D) &= \sum_t \Pr_\theta(C = \text{no}, T = t, I = \text{yes}) \\ &= \sum_t \theta_{C=\text{no}} \cdot \theta_{T=t|C=\text{no}} \cdot \theta_{I=\text{yes}|T=t} \\ &= \theta_{C=\text{no}} \cdot \Big(\theta_{-ve|\text{no}}\, \theta_{\text{yes}|-ve} + \theta_{+ve|\text{no}}\, \theta_{\text{yes}|+ve}\Big), \end{aligned}$$

abbreviating $\theta_{t|\text{no}} = \theta_{T=t|C=\text{no}}$ and $\theta_{\text{yes}|t} = \theta_{I=\text{yes}|T=t}$. Call the bracket $S(\theta)$; it is a convex combination of $\theta_{\text{yes}|-ve}, \theta_{\text{yes}|+ve} \in [0,1]$, so $L(\theta \mid D) \le 1$ with the bound attained at $\theta_{C=\text{no}} = \theta_{\text{yes}|-ve} = \theta_{\text{yes}|+ve} = 1$. Hence θ is ML iff $\theta_{C=\text{no}} = 1$ — the first bullet — and $S(\theta) = 1$. Writing $\lambda = \theta_{-ve|\text{no}}$, so that $S(\theta) = \lambda\theta_{\text{yes}|-ve} + (1-\lambda)\theta_{\text{yes}|+ve} \le 1$, equality holds iff $\lambda(1-\theta_{\text{yes}|-ve}) = 0$ and $(1-\lambda)(1-\theta_{\text{yes}|+ve}) = 0$, that is, iff $\theta_{\text{yes}|t} = 1$ for every t with $\theta_{t|\text{no}} > 0$:

- (i) $\lambda = 1$: forces $\theta_{I=\text{yes}|T=-ve} = 1$, the first listed alternative.
- (ii) $\lambda = 0$: forces $\theta_{I=\text{yes}|T=+ve} = 1$, the second alternative.
- (iii) $0 < \lambda < 1$: forces both, the third alternative, with $\theta_{T|C=\text{no}}$ unconstrained.

Conversely each case gives $S(\theta) = 1$ (Check!), so with $\theta_{C=\text{no}} = 1$ each attains $L(\theta \mid D) = 1$.

Problem 17.19. Consider the data set D in Exercise 17.1 and the following network structures:

- G_1: $A \to B$, $B \to C$, $C \to D$, and $A \to D$
- G_2: $A \to B$, $B \to C$, $C \to D$, and $B \to D$.

The data set D of Exercise 17.1 is the following complete data set over the binary variables A, B, C, D (values T and F):

Case	A	B	C	D
1	T	F	F	F
2	T	F	F	T
3	F	F	T	F
4	T	T	F	T
5	F	F	T	T
6	F	T	T	F
7	F	T	T	T
8	T	F	F	T
9	F	F	T	F
10	T	T	T	T

Which structure has a higher likelihood given D? What is the exact value for $LL(G_1 \mid D) - LL(G_2 \mid D)$?

Solution. G_1 has the higher likelihood, by $LL(G_1 \mid D) - LL(G_2 \mid D) = 15\log 3 - 5\log 5 - 12 \approx 0.165$ (logarithms base 2, $N = 10$).

By Theorem 17.3, $LL(G \mid D) = -N \sum_{XU} ENT_D(X \mid U)$ over the families of G, which are

$$\begin{aligned} G_1: &\quad A,\ BA,\ CB,\ DCA, \\ G_2: &\quad A,\ BA,\ CB,\ DCB. \end{aligned}$$

They differ only in the family of D, so all other entropy terms cancel and

$$LL(G_1 \mid D) - LL(G_2 \mid D) = -N\big(ENT_D(D \mid C, A) - ENT_D(D \mid C, B)\big),$$

each family score computed from counts as $-N\, ENT_D(X \mid U) = \sum_{xu} D\#(xu) \log \frac{D\#(xu)}{D\#(u)}$. For DCA:

C	A	$D\#(C,A)$	$D=F$	$D=T$	cases
F	T	4	1	3	1, 2, 4, 8
T	F	5	3	2	3, 5, 6, 7, 9
T	T	1	0	1	10
F	F	0	0	0	none

(The instantiation $C = F, A = F$ never occurs and contributes nothing, by Exercise 17.3.) Hence

$$\begin{aligned} -N \cdot ENT_D(D \mid C, A) &= 1 \log \tfrac{1}{4} + 3 \log \tfrac{3}{4} + 3 \log \tfrac{3}{5} + 2 \log \tfrac{2}{5} + 1 \log 1 \\ &= -2 + 3(\log 3 - 2) + 3(\log 3 - \log 5) + 2(1 - \log 5) \\ &= -6 + 6 \log 3 - 5 \log 5. \end{aligned}$$

For DCB:

C	B	$D\#(C,B)$	$D=F$	$D=T$	cases
F	F	3	1	2	1, 2, 8
F	T	1	0	1	4
T	F	3	2	1	3, 5, 9
T	T	3	1	2	6, 7, 10

The three parent instantiations with three cases each split 1–2 or 2–1, so each contributes $1 \log \frac{1}{3} + 2 \log \frac{2}{3}$, while $C = F, B = T$ contributes $1 \log 1 = 0$. Hence

$$\begin{aligned} -N \cdot ENT_D(D \mid C, B) &= 3\Big(1 \log \tfrac{1}{3} + 2 \log \tfrac{2}{3}\Big) \\ &= 3\big(-\log 3 + 2(1 - \log 3)\big) \\ &= 6 - 9 \log 3. \end{aligned}$$

Subtracting,

$$\begin{aligned} LL(G_1 \mid D) - LL(G_2 \mid D) &= \big(-6 + 6 \log 3 - 5 \log 5\big) - \big(6 - 9 \log 3\big) \\ &= 15 \log 3 - 5 \log 5 - 12. \end{aligned}$$

which is $\log \frac{3^{15}}{5^5 \cdot 2^{12}} = \log \frac{14348907}{12800000} \approx 0.165 > 0$, so G_1 wins.

Problem 17.20. Compute the MDL score for the network structure in Figure 17.10(c) given the data set in (17.11).

Figure 17.10(c) is the complete DAG over the four variables A, B, C, D obtained from the tree of Figure 17.10(a) by adding all remaining edges consistent with the variable order B, D, C, A. Its edges are

$$B \to D, \quad B \to C, \quad B \to A, \quad D \to C, \quad D \to A, \quad C \to A,$$

so the families are B, DB, CBD and $ABCD$. All four variables are binary.

The data set (17.11) is the following complete data set of size $N = 5$:

Case	A	B	C	D
1	a1	b1	c2	d1
2	a1	b1	c2	d2
3	a1	b2	c1	d1
4	a2	b1	c1	d2
5	a1	b1	c2	d2

Recall from (17.14) that the MDL score of a structure G given a data set D of size N is

$$MDL(G \mid D) \;=\; LL(G \mid D) - \frac{\log_2 N}{2}\,\|G\|,$$

where $\|G\|$ is the dimension of G, that is, the number of independent network parameters.

Solution. $MDL(G \mid D) = 2 - \frac{25}{2}\log_2 5 \approx -27.0$, from $LL(G \mid D) = 2 - 5\log 5$ and $\|G\| = 15$ at $N = 5$ (logarithms base 2 throughout). By the definition of dimension,

$$\|G\| = \sum_i \|X_i \mathbf{U}_i\|, \qquad \|X_i \mathbf{U}_i\| = (X_i\# - 1)\,\mathbf{U}_i\#,$$

where $X_i\#$ and $\mathbf{U}_i\#$ are the numbers of instantiations of X_i and of its parents $\mathbf{U}_i$. With all four variables binary and families B, DB, CBD, $ABCD$:

Family	Parents	$(X\# - 1)$	$\mathbf{U}\#$	Dimension
B	none	1	1	1
DB	B	1	2	2
CBD	B, D	1	4	4
$ABCD$	B, C, D	1	8	8

Hence $\|G\| = 1 + 2 + 4 + 8 = 15$. Since G is a complete DAG its ML parametrization induces exactly $\Pr_D$ (Exercise 17.8), so

$$LL(G \mid D) \;=\; \sum_{i=1}^{N} \log \Pr_D(d_i) \;=\; \sum_{\alpha} D\#(\alpha) \log \frac{D\#(\alpha)}{N},$$

α ranging over complete instantiations. Cases 2 and 5 coincide, the rest are distinct:

Instantiation	Count	$\Pr_D$
$a1, b1, c2, d1$	1	1/5
$a1, b1, c2, d2$	2	2/5
$a1, b2, c1, d1$	1	1/5
$a2, b1, c1, d2$	1	1/5

Hence

$$\begin{aligned} LL(G \mid D) &= 3\log\tfrac{1}{5} + 2\log\tfrac{2}{5} \\ &= -3\log 5 + 2\big(1 - \log 5\big) \\ &= 2 - 5\log 5 \;\approx\; 2 - 11.6096 \;=\; -9.61. \end{aligned}$$

Method (2): decomposing over the families of G by Theorem 17.3,

$$LL(G \mid D) = -N\big(ENT_D(B) + ENT_D(D \mid B) + ENT_D(C \mid B, D) + ENT_D(A \mid B, C, D)\big).$$

Computing each term from counts:

- B: $b1$ occurs 4 times, $b2$ once, so $-N \cdot ENT_D(B) = 4\log\frac{4}{5} + \log\frac{1}{5} = 8 - 5\log 5 \approx -3.61$.
- $D \mid B$: among the four $b1$ cases, $d1$ once and $d2$ three times; the single $b2$ case has $d1$. So $-N \cdot ENT_D(D \mid B) = \log\frac{1}{4} + 3\log\frac{3}{4} = -8 + 3\log 3 \approx -3.25$.
- $C \mid B, D$: the only parent instantiation with more than one case is $b1, d2$ (cases $2, 4, 5$), where $c2$ occurs twice and $c1$ once. So $-N \cdot ENT_D(C \mid B, D) = 2\log\frac{2}{3} + \log\frac{1}{3} = 2 - 3\log 3 \approx -2.75$.
- $A \mid B, C, D$: within every parent instantiation of A the value of A is constant (cases 2 and 5 agree completely), so all conditional probabilities are 1 and $-N \cdot ENT_D(A \mid B, C, D) = 0$.

Adding, $LL(G \mid D) = (8 - 5\log 5) + (-8 + 3\log 3) + (2 - 3\log 3) = 2 - 5\log 5$ as before. With penalty weight $\frac{\log_2 5}{2} = 1.16096$,

$$\begin{aligned} MDL(G \mid D) &= LL(G \mid D) - \frac{\log_2 5}{2}\,\|G\| \\ &= \left(2 - 5\log_2 5\right) - \tfrac{15}{2}\log_2 5 \\ &= 2 - \tfrac{25}{2}\log_2 5 \\ &\approx -9.61 - 17.41 \\ &= -27.02. \end{aligned}$$

Against the other two structures of Figure 17.10, scored in Section 17.4.3:

Structure	$\|G\|$	$LL(G \mid D)$	Penalty	MDL
(a) tree	7	-12.1	8.1	-20.2
(b) DAG with $D \to A$	9	-10.1	10.4	-20.5
(c) complete DAG	15	-9.6	17.4	-27.0

The complete DAG has the largest log-likelihood, as Theorem 17.11 requires, and the worst MDL score.

17 Learning: The Bayesian Approach

Contents

17.1 Exercises 18.1–18.7 308
17.2 Exercises 18.8–18.14 314
17.3 Exercises 18.15–18.21 319
17.4 Exercises 18.22–18.24 326

17.1 Exercises 18.1–18.7

Problem 18.1. Consider the meta-network in Figure 18.2 and the corresponding data set D.
The base network of Figure 18.1 has three binary variables: H (health aware), S (smokes), E (exercises regularly), with edges $H \to S$ and $H \to E$. The meta-network of Figure 18.2 is built from N instances of this structure, the ith instance containing H_i, S_i, E_i with edges $H_i \to S_i$ and $H_i \to E_i$. Three parameter-set nodes are added, namely θ_H, $\theta_{S|\bar{h}}$ and $\theta_{E|\bar{h}}$, together with the edges $\theta_H \to H_i$, $\theta_{S|\bar{h}} \to S_i$ and $\theta_{E|\bar{h}} \to E_i$ for every $i = 1, \ldots, N$. The parameter sets $\theta_{S|h}$ and $\theta_{E|h}$ do not appear as nodes because their values are fixed.
The data set D is

Case	H	S	E
1	F	F	T
2	T	F	T
3	T	F	T
4	F	F	F
5	F	T	F

Assume the following fixed values of parameter sets:

$$\theta_{S|h} = (.1, .9), \qquad \theta_{E|h} = (.8, .2).$$

For the remaining sets, suppose that we have the following prior distributions:

$\theta_H = (\theta_h, \theta_{\bar{h}})$	$P(\theta_H)$
$(.75, .25)$	80%
$(.90, .10)$	20%
$\theta_{S|\bar{h}} = (\theta_{s|\bar{h}}, \theta_{\bar{s}|\bar{h}})$	$P(\theta_{S|\bar{h}})$
$(.25, .75)$	10%
$(.50, .50)$	90%
$\theta_{E|\bar{h}} = (\theta_{e|\bar{h}}, \theta_{\bar{e}|\bar{h}})$	$P(\theta_{E|\bar{h}})$
$(.50, .50)$	35%
$(.75, .25)$	65%

Compute the following given the data set D:
- (a) The maximum likelihood parameter estimates.
- (b) The MAP parameter estimates.
- (c) The Bayesian parameter estimates.

Compute the probability of observing an individual who is a smoker and exercises regularly given each of the above parameter estimates. Compute also $P(S_6 = s, E_6 = e \mid D)$ with respect to the meta-network.

Solution. Since D is complete, the likelihood (18.7), the prior (18.5) and the posterior (18.6) all factorize over the free parameter sets θ_H, $\theta_{S|\bar{h}}$, $\theta_{E|\bar{h}}$, each of which is therefore maximized on its own; $\theta_{S|h}$ and $\theta_{E|h}$ are fixed and play no part. The counts are (writing h for $H = T$, s for $S = T$, e for $E = T$)

$$\begin{aligned} D\#(h) &= 2, & D\#(\bar{h}) &= 3, \\ D\#(s, \bar{h}) &= 1, & D\#(\bar{s}, \bar{h}) &= 2, \\ D\#(e, \bar{h}) &= 1, & D\#(\bar{e}, \bar{h}) &= 2. \end{aligned}$$

(a) Each set maximizes its own likelihood factor $\prod_x \theta_{x|u}^{D\#(xu)}$:

$$\begin{aligned} \theta_H &: \ (.75)^2(.25)^3 = .0087890625 \ > \ (.90)^2(.10)^3 = .00081, \\ \theta_{S|\bar{h}} &: \ (.25)(.75)^2 = .140625 \ > \ (.50)(.50)^2 = .125, \\ \theta_{E|\bar{h}} &: \ (.50)(.50)^2 = .125 \ > \ (.75)(.25)^2 = .046875, \end{aligned}$$

so θ^{ml} takes $\theta_H = (.75, .25)$, $\theta_{S|\bar{h}} = (.25, .75)$ and $\theta_{E|\bar{h}} = (.50, .50)$.

(b) By Theorem 18.4 each posterior is proportional to prior times the factor above:

$$\begin{aligned}\theta_H &: \ .80(.0087890625) = .00703125 \ > \ .20(.00081) = .000162,\\ \theta_{S|\bar{h}} &: \ .10(.140625) = .0140625 \ < \ .90(.125) = .1125,\\ \theta_{E|\bar{h}} &: \ .35(.125) = .04375 \ > \ .65(.046875) = .03046875,\end{aligned}$$

so θ^{ma} takes $\theta_H = (.75, .25)$, $\theta_{S|\bar{h}} = (.50, .50)$ and $\theta_{E|\bar{h}} = (.50, .50)$, the prior having flipped $\theta_{S|\bar{h}}$ relative to maximum likelihood.
(c) Normalizing those three lists gives the posteriors of Theorem 18.4,

$$\begin{aligned}P(\theta_H = (.75, .25) \mid D) &= \tfrac{3125}{3197} \approx 97.75\%,\\ P(\theta_{S|\bar{h}} = (.25, .75) \mid D) &= \tfrac{1}{9} \approx 11.11\%,\\ P(\theta_{E|\bar{h}} = (.50, .50) \mid D) &= \tfrac{56}{95} \approx 58.95\%,\end{aligned}$$

the alternatives taking the complements, and the Bayesian estimates are their expectations (Definition 18.3):

$$\begin{aligned}\theta^{be}_{h} &= .75 \times \tfrac{3125}{3197} + .90 \times \tfrac{72}{3197} \approx .7534,\\ \theta^{be}_{s|\bar{h}} &= .25 \times \tfrac{1}{9} + .50 \times \tfrac{8}{9} = \tfrac{17}{36} \approx .4722,\\ \theta^{be}_{e|\bar{h}} &= .50 \times \tfrac{56}{95} + .75 \times \tfrac{39}{95} = \tfrac{229}{380} \approx .6026,\end{aligned}$$

with $\theta^{be}_{\bar{h}} \approx .2466$, $\theta^{be}_{\bar{s}|\bar{h}} \approx .5278$, $\theta^{be}_{\bar{e}|\bar{h}} \approx .3974$, and the fixed sets keeping their values (their posteriors being point masses).
In the base network $S \leftarrow H \rightarrow E$ we have $Pr_\theta(s, e) = \theta_h \theta_{s|h} \theta_{e|h} + \theta_{\bar{h}} \theta_{s|\bar{h}} \theta_{e|\bar{h}}$, with $\theta_{s|h} = .1$ and $\theta_{e|h} = .8$ throughout, so

$$\begin{aligned}Pr_{\theta^{ml}}(s, e) &= (.75)(.1)(.8) + (.25)(.25)(.50) = .09125,\\ Pr_{\theta^{ma}}(s, e) &= (.75)(.1)(.8) + (.25)(.50)(.50) = .1225,\\ Pr_{\theta^{be}}(s, e) &\approx (.7534)(.1)(.8) + (.2466)(.4722)(.6026) \approx .13045.\end{aligned}$$

The data set being complete, Theorem 18.3 with α the event $S = s, E = e$ gives

$$P(S_6 = s, E_6 = e \mid D) = Pr_{\theta^{be}}(s, e) \approx 13.05\%,$$

which is what summing $Pr_\theta(s, e) P(\theta \mid D)$ over the eight parametrizations, as Theorem 18.2 prescribes, returns as well.

Problem 18.2. Consider a network structure $A \rightarrow B$ where variables A and B are binary, with values a_1, a_2 and b_1, b_2. Suppose that we have the following priors on parameter sets:

$\theta_A = (\theta_{a_1}, \theta_{a_2})$	$P(\theta_A)$
$(.80, .20)$	70%
$(.20, .80)$	30%

$\theta_{B\|a_1} = (\theta_{b_1\|a_1}, \theta_{b_2\|a_1})$	$P(\theta_{B\|a_1})$
$(.90, .10)$	50%
$(.10, .90)$	50%

$\theta_{B\|a_2} = (\theta_{b_1\|a_2}, \theta_{b_2\|a_2})$	$P(\theta_{B\|a_2})$
$(.95, .05)$	60%
$(.05, .95)$	40%

Compute the posterior distributions of parameter sets given the data set

D	A	B
d_1	a_1	b_2
d_2	a_1	b_2
d_3	a_1	b_1
d_4	a_2	b_2

Compute also the expected parameter values given this data set. Finally, evaluate the query $P(A_5 = a_1 \mid B_5 = b_2, D)$.

Solution. The data set is complete, so Theorem 18.1 gives parameter independence given D and Theorem 18.4 gives each posterior in closed form,

$$P(\theta_{X|u} \mid D) = \eta\, P(\theta_{X|u}) \prod_x \theta_{x|u}^{D\#(xu)}.$$

With counts $D\#(a_1) = 3$, $D\#(a_2) = 1$, $D\#(b_1a_1) = 1$, $D\#(b_2a_1) = 2$, $D\#(b_1a_2) = 0$ and $D\#(b_2a_2) = 1$, prior times likelihood term gives

$$\begin{aligned}
\theta_A &: \ .70(.80)^3(.20) = .07168, \quad .30(.20)^3(.80) = .00192,\\
\theta_{B|a_1} &: \ .50(.90)(.10)^2 = .0045, \quad .50(.10)(.90)^2 = .0405,\\
\theta_{B|a_2} &: \ .60(.05) = .03, \quad .40(.95) = .38,
\end{aligned}$$

and normalizing by .0736, .045 and .41 respectively,

$$\begin{aligned}
P(\theta_A = (.80,.20) \mid D) &= \tfrac{112}{115} \approx 97.39\%, & P(\theta_A = (.20,.80) \mid D) &= \tfrac{3}{115},\\
P(\theta_{B|a_1} = (.90,.10) \mid D) &= 10\%, & P(\theta_{B|a_1} = (.10,.90) \mid D) &= 90\%,\\
P(\theta_{B|a_2} = (.95,.05) \mid D) &= \tfrac{3}{41}, & P(\theta_{B|a_2} = (.05,.95) \mid D) &= \tfrac{38}{41},
\end{aligned}$$

the joint posterior being the product of the three (Theorem 18.1). The expected parameter values are the Bayesian estimates of Definition 18.3:

$$\begin{aligned}
\theta^{be}_{a_1} &= .80 \times \tfrac{112}{115} + .20 \times \tfrac{3}{115} = \tfrac{451}{575} \approx .7843,\\
\theta^{be}_{b_1|a_1} &= .90 \times .10 + .10 \times .90 = .18,\\
\theta^{be}_{b_1|a_2} &= .95 \times \tfrac{3}{41} + .05 \times \tfrac{38}{41} = \tfrac{19}{164} \approx .1159,
\end{aligned}$$

with $\theta^{be}_{a_2} = \frac{124}{575}$, $\theta^{be}_{b_2|a_1} = .82$ and $\theta^{be}_{b_2|a_2} = \frac{145}{164}$. Completeness of D lets Theorem 18.3, in the conditional form of Exercise 18.7, answer the query by inference on the base network $A \to B$ at θ^{be}:

$$\begin{aligned}
P(A_5 = a_1 \mid B_5 = b_2, D) = Pr_{\theta^{be}}(a_1 \mid b_2) &= \frac{\theta^{be}_{a_1}\theta^{be}_{b_2|a_1}}{\theta^{be}_{a_1}\theta^{be}_{b_2|a_1} + \theta^{be}_{a_2}\theta^{be}_{b_2|a_2}}\\
&= \frac{.64317}{.64317 + .19066} = \tfrac{758131}{982881} \approx 77.13\%.
\end{aligned}$$

Problem 18.3. Consider Exercise 18.2, which concerns the network structure $A \to B$ with binary variables A (values a_1, a_2) and B (values b_1, b_2), and the parameter sets

$$\begin{aligned}
\theta_A &\in \{(.80,.20),(.20,.80)\},\\
\theta_{B|a_1} &\in \{(.90,.10),(.10,.90)\},\\
\theta_{B|a_2} &\in \{(.95,.05),(.05,.95)\}.
\end{aligned}$$

Describe a meta-network that captures the following constraint

$$\theta_{B|a_1} = (.90,.10) \ \text{ iff } \ \theta_{B|a_2} = (.05,.95).$$

Solution. Add the single edge $\theta_{B|a_1} \to \theta_{B|a_2}$ to the meta-network of Exercise 18.2 and make $\theta_{B|a_2}$ a deterministic function of its parent. A meta-network as given by Definition 18.2 cannot express the constraint — its parameter-set nodes are all roots, hence mutually independent (Theorem 18.1) — but Section 18.2.3 permits exactly this relaxation.

So keep the instance variables $A_i \to B_i$ and the edges $\theta_A \to A_i$, $\theta_{B|a_1} \to B_i$, $\theta_{B|a_2} \to B_i$ for $i = 1, \ldots, N$; the node $\theta_{B|a_1}$ retains its prior and $\theta_{B|a_2}$ gets the CPT

$\theta_{B\|a_1}$	$\theta_{B\|a_2}$	$P(\theta_{B\|a_2} \mid \theta_{B\|a_1})$
$(.90, .10)$	$(.95, .05)$	0
$(.90, .10)$	$(.05, .95)$	1
$(.10, .90)$	$(.95, .05)$	1
$(.10, .90)$	$(.05, .95)$	0

whose two zero rows encode the "only if" and the "if" halves of the biconditional. The rest, including the instance CPTs (18.1), is unchanged: $P(B_i \mid A_i = a_k, \theta_{B|a_1}, \theta_{B|a_2}) = \theta_{B|a_k}$.
The priors of Exercise 18.2 cannot both survive: the constraint forces $P(\theta_{B|a_1} = (.90, .10)) = P(\theta_{B|a_2} = (.05, .95))$, which the given 50% and 40% violate. Keeping $P(\theta_{B|a_1}) = (50\%, 50\%)$ induces $P(\theta_{B|a_2} = (.05, .95)) = 50\%$.
Method (2): introduce a binary root T with $T \to \theta_{B|a_1}$ and $T \to \theta_{B|a_2}$, where $T = t$ stands for the hypothesis "$\theta_{B|a_1} = (.90, .10)$ and $\theta_{B|a_2} = (.05, .95)$" and $T = \bar{t}$ for its complement, both CPTs being deterministic:

$$P(\theta_{B|a_1} = (.90, .10) \mid t) = 1, \quad P(\theta_{B|a_1} = (.10, .90) \mid \bar{t}) = 1,$$
$$P(\theta_{B|a_2} = (.05, .95) \mid t) = 1, \quad P(\theta_{B|a_2} = (.95, .05) \mid \bar{t}) = 1.$$

Both dependencies being functional, this is the same distribution as before with $P(t) = P(\theta_{B|a_1} = (.90, .10))$.

Problem 18.4. Consider a Bayesian network structure with three binary variables A, B, and C and two edges $A \to B$ and $A \to C$. Suppose that the CPTs for variables B and C are equal and these CPTs are independent of the one for variable A. Draw the structure of a meta-network for this situation. Develop a closed form for computing the posterior distribution on parameter set $\theta_{B|a_1}$ given a complete data set D. Assume first that parameter sets are discrete, deriving a closed form that resembles Equation 18.4. Then generalize the form to continuous parameter sets with Dirichlet distributions.
Here Equation 18.4 is the statement of Theorem 18.4: for a discrete parameter set $\theta_{X|u}$ and a complete data set D,

$$P(\theta_{X|u} \mid D) = \eta \, P(\theta_{X|u}) \prod_x \theta_{x|u}^{D\#(xu)},$$

where η is a normalizing constant.

Solution. Tying the two CPTs pools the counts of the two children, and the closed form becomes

$$P(\theta_{B|a_1} \mid D) = \eta \, P(\theta_{B|a_1}) \prod_j \theta_{b_j|a_1}^{D\#(b_j a_1) + D\#(c_j a_1)},$$

that is, (18.4) with $D\#(b\,a_1)$ replaced by the pooled count $M\#(b\,a_1)$ defined below. Here $b_j \leftrightarrow c_j$ is the correspondence under which the two CPTs are equal, so that the assumption reads $\theta_{c_j|a_k} = \theta_{b_j|a_k}$ and the model has three distinct parameter sets, not five: θ_A, $\theta_{B|a_1}$, $\theta_{B|a_2}$, the last two doing double duty as the parameters of C.
The meta-network is N instances $A_i \to B_i$, $A_i \to C_i$ beneath three parameter roots, with edges

$$\theta_A \to A_i, \qquad \theta_{B|a_1} \to B_i,\ \theta_{B|a_1} \to C_i, \qquad \theta_{B|a_2} \to B_i,\ \theta_{B|a_2} \to C_i$$

for $i = 1, \ldots, N$, so every B_i and every C_i has the three parents A_i, $\theta_{B|a_1}$, $\theta_{B|a_2}$. This departs from Definition 18.2 in one respect only, one parameter node per parent instantiation feeding both children instead of separate nodes for the families $B \mid A$ and $C \mid A$; every parameter node is still a root. The instance CPTs follow (18.1) with the shared sets, $P(B_i \mid A_i = a_k, \theta) = P(C_i \mid A_i = a_k, \theta) = \theta_{B|a_k}$, the vector $\theta_{B|a_k}$ read in the second case as assigning $\theta_{b_j|a_k}$ to $C_i = c_j$.
Parameter independence survives: instantiating every A_i, B_i, C_i to its value in D prunes the edges out of each observed A_i (Section 6.9.2) and makes $\theta_{B|a_k} \to B_i$, $\theta_{B|a_k} \to C_i$ prunable whenever $A_i \neq a_k$,

leaving θ_A, $\theta_{B|a_1}$, $\theta_{B|a_2}$ with no common neighbours, hence pairwise d-separated by the evidence, so $P(\theta \mid D)$ factorizes as in Theorem 18.1.
For the closed form, let D be complete of size N with cases $d_i = (a^i, b^i, c^i)$. By (18.7), and since every factor $\theta_{c^i|a^i}$ is again one of the $\theta_{b_j|a_k}$,

$$P(D \mid \theta) = \prod_{i=1}^{N} \theta_{a^i}\, \theta_{b^i|a^i}\, \theta_{c^i|a^i} = \prod_k \theta_{a_k}^{D\#(a_k)} \prod_k \prod_j \theta_{b_j|a_k}^{M\#(b_j a_k)},$$

where $M\#(b_j a_k) \stackrel{\text{def}}{=} D\#(b_j a_k) + D\#(c_j a_k)$ counts both children's observations of the shared distribution for parent value a_k; note $\sum_j M\#(b_j a_k) = 2D\#(a_k)$. Multiplying by the factorized prior $P(\theta_A)P(\theta_{B|a_1})P(\theta_{B|a_2})$ and summing out θ_A and $\theta_{B|a_2}$ leaves the marginal displayed at the outset; $\theta_{B|a_2}$ obeys the same form, θ_A obeys (18.4) unchanged.
For continuous sets, a Dirichlet prior $\rho(\theta_{B|a_1}) = \eta_0 \prod_b \theta_{b|a_1}^{\psi_{b|a_1}-1}$ (Definition 18.5) meets the same likelihood factor, which does not depend on discreteness, so

$$\rho(\theta_{B|a_1} \mid D) = \eta\, \rho(\theta_{B|a_1}) \prod_b \theta_{b|a_1}^{M\#(b\,a_1)} = \eta' \prod_b \theta_{b|a_1}^{\psi_{b|a_1}+M\#(b\,a_1)-1},$$

again Dirichlet, with exponents $\psi'_{b_j|a_1} = \psi_{b_j|a_1} + D\#(b_j a_1) + D\#(c_j a_1)$ — the analogue of (18.15) in Theorem 18.9 with pooled counts — and equivalent sample size $\psi'_{B|a_1} = \psi_{B|a_1} + 2D\#(a_1)$. The Dirichlet expectation (18.10) then gives the analogue of (18.16),

$$\theta^{be}_{b_j|a_1} = \frac{\psi_{b_j|a_1} + D\#(b_j\, a_1) + D\#(c_j\, a_1)}{\psi_{B|a_1} + 2\, D\#(a_1)}.$$

Problem 18.5. Consider a meta-network as given by Definition 18.2. Let Σ contain X_N for every variable X in structure G and let Γ contain $X_1, \ldots, X_{N-1}$ for every variable X in structure G. Show that Σ is d-separated from Γ by the nodes representing parameter sets.
Recall Definition 18.2: a meta-network of size N for structure G is constructed from N instances of G, with variable X of G appearing as X_i in the ith instance; and for every variable X of G and parent instantiation u, the meta-network contains a node $\theta_{X|u}$ together with the edges $\theta_{X|u} \to X_1, \ldots, \theta_{X|u} \to X_N$.

Solution. Every path between Σ and Γ passes through a parameter node as a divergent valve, which the set Θ of all parameter-set nodes $\theta_{X|u}$ closes.
By Definition 18.2 the meta-network's nodes split into Θ and the instance sets $V_i = \{X_i : X \text{ a variable of } G\}$, and its edges are of two kinds only: intra-instance edges $U_i \to X_i$, one for each edge $U \to X$ of G and each i, and parameter edges $\theta_{X|u} \to X_i$. Hence no edge joins V_i to V_j for $i \neq j$. Let P be a path (a sequence of distinct nodes, consecutive ones adjacent in either direction) from $X_N \in \Sigma \subseteq V_N$ to $Y_j \in \Gamma \subseteq V_j$ with $j < N$, and let W be the first node on P outside V_N, which exists because $Y_j \notin V_N$. Its predecessor lies in V_N, so W is adjacent to a node of V_N and is therefore no instance node at all: $W \in \Theta$. Nor is W an endpoint, the endpoints lying in Σ and Γ, which are disjoint from Θ; so P contains a subpath $W' \leftarrow W \to W''$, both edges directed away from W because Definition 18.2 gives a parameter node only outgoing edges. This divergent valve is closed by $W \in \Theta$, so P is blocked, and as P was arbitrary,

$$d\text{-}sep(\Sigma, \Theta, \Gamma)$$

holds; by soundness of d-separation, Σ and Γ are independent given Θ.

Problem 18.6. Consider Theorem 18.2 and show that $Pr_\theta(\alpha) = P(\alpha_i \mid \theta)$, where α_i results from replacing every variable X in α by its instance X_i.
Here P is the meta-distribution induced by a meta-network as given by Definition 18.2, θ is an instantiation of all its parameter-set nodes (a parametrization of the base structure G), and Pr_θ is the base distribution induced by G under θ. Theorem 18.2 states that, given discrete parameter sets and a data set D of size N,

$$P(\alpha_{N+1} \mid D) = \sum_\theta Pr_\theta(\alpha) P(\theta \mid D).$$

Solution. Given θ, the N instances are independent copies of the base network, so instance i carries exactly the distribution Pr_θ.
Fix θ with $P(\theta) > 0$, and for an instantiation $\mathbf{x}$ of G's variables $X^{(1)}, \ldots, X^{(n)}$ write $\mathbf{x}^i$ for the corresponding instantiation of $V_i = \{X_i^{(1)}, \ldots, X_i^{(n)}\}$; the map $\mathbf{x} \mapsto \mathbf{x}^i$ is a bijection. The meta-network is a Bayesian network, so by the chain rule (Chapter 3),

$$P(\theta, \mathbf{x}^1, \ldots, \mathbf{x}^N) = \Big[\prod_{X,u} P(\theta_{X|u})\Big] \prod_{i=1}^{N} \prod_X P(x_i \mid u_i, \theta),$$

with x_i and u_i the values of X_i and its instance parents in $\mathbf{x}^i$. The bracketed factor is $P(\theta)$, the parameter nodes being roots, and $P(x_i \mid u_i, \theta) = \theta_{x|u}$ by the semantics (18.1) of instance CPTs, so dividing by $P(\theta)$,

$$P(\mathbf{x}^1, \ldots, \mathbf{x}^N \mid \theta) = \prod_{i=1}^{N} \prod_X \theta_{x|u} = \prod_{i=1}^{N} Pr_\theta(\mathbf{x}^i),$$

the last equality being the chain rule for the base network (G, θ). Summing out the instances other than i,

$$P(\mathbf{x}^i \mid \theta) = Pr_\theta(\mathbf{x}^i) \prod_{j \neq i} \Big[\sum_{\mathbf{x}^j} Pr_\theta(\mathbf{x}^j)\Big] = Pr_\theta(\mathbf{x}^i),$$

each bracketed sum being 1. Finally, the substitution $X \mapsto X_i$ is uniform and commutes with the Boolean connectives, so $\mathbf{x} \models \alpha$ iff $\mathbf{x}^i \models \alpha_i$, and

$$P(\alpha_i \mid \theta) = \sum_{\mathbf{x}^i \models \alpha_i} P(\mathbf{x}^i \mid \theta) = \sum_{\mathbf{x} \models \alpha} Pr_\theta(\mathbf{x}) = Pr_\theta(\alpha).$$

Problem 18.7. Consider Theorem 18.3 and show that

$$P(\alpha_{N+1} \mid \beta_{N+1}, D) = Pr_{\theta^{be}}(\alpha \mid \beta).$$

Theorem 18.3 states that, given discrete parameter sets and a complete data set D of size N,

$$P(\gamma_{N+1} \mid D) = Pr_{\theta^{be}}(\gamma),$$

where γ_{N+1} is obtained from the event γ by replacing every variable X by its instance X_{N+1}, and θ^{be} are the Bayesian estimates of Definition 18.3 given D.

Solution. The substitution $X \mapsto X_{N+1}$ leaves the Boolean structure of a sentence untouched, so $(\alpha \wedge \beta)_{N+1} = \alpha_{N+1} \wedge \beta_{N+1}$, with $\alpha \wedge \beta$ again an event over the base variables. The data set D is complete, the hypothesis of Theorem 18.3, so applying (18.3) first with $\gamma = \alpha \wedge \beta$ and then with $\gamma = \beta$,

$$\begin{aligned} P(\alpha_{N+1} \mid \beta_{N+1}, D) &= \frac{P\big((\alpha \wedge \beta)_{N+1} \mid D\big)}{P(\beta_{N+1} \mid D)} = \frac{Pr_{\theta^{be}}(\alpha \wedge \beta)}{Pr_{\theta^{be}}(\beta)} \\ &= Pr_{\theta^{be}}(\alpha \mid \beta), \end{aligned}$$

as required; the standing assumption $P(\beta_{N+1} \mid D) > 0$ makes the left side defined, and equals $Pr_{\theta^{be}}(\beta)$ by the second application, so the right side is defined too.

17.2 Exercises 18.8–18.14

Problem 18.8. Consider Theorem 18.2 and show that

$$P(\alpha_{N+1} \mid \beta_{N+1}, D) = \sum_{\theta} Pr_{\theta}(\alpha \mid \beta)\, P(\theta \mid \beta_{N+1}, D).$$

Here the meta-network is the one of Definition 18.2: it contains a node for every discrete parameter set $\theta_{X|u}$ of the base structure G, together with $N+1$ instances $X_1, \ldots, X_{N+1}$ of each base variable X and the edges $\theta_{X|u} \to X_i$; the data set D of size N is asserted as evidence on instances $1, \ldots, N$. As in Theorem 18.2, the event α_{N+1} is obtained from the base-network event α by replacing every occurrence of a variable X by its instance X_{N+1}, and similarly for β_{N+1}. Assume $P(\beta_{N+1} \mid \theta) > 0$ for every parametrization θ with $P(\theta \mid D) > 0$.

Solution. Case analysis on θ (the parametrizations are mutually exclusive and exhaustive) gives

$$P(\alpha_{N+1} \mid \beta_{N+1}, D) = \sum_{\theta} P(\alpha_{N+1} \mid \beta_{N+1}, D, \theta)\, P(\theta \mid \beta_{N+1}, D),$$

so it suffices that $P(\alpha_{N+1} \mid \beta_{N+1}, D, \theta) = Pr_{\theta}(\alpha \mid \beta)$.
(i) *The data drop out.* With Δ the instance-$(N+1)$ variables, Γ those of instances $1, \ldots, N$, and Θ the parameter nodes, Exercise 18.5 gives *d-sep*(Δ, Θ, Γ) – the edges $\theta_{X|u} \to X_i$ are the only ones joining instances, each a diverging valve at a root closed by instantiating Θ – hence $I_P(\Delta, \Theta, \Gamma)$ by soundness of d-separation (Theorem 4.2). Applying it to $\alpha_{N+1} \wedge \beta_{N+1}$ and to β_{N+1}, both events over Δ while D is over Γ, and dividing, $P(\alpha_{N+1} \mid \beta_{N+1}, D, \theta) = P(\alpha_{N+1} \mid \beta_{N+1}, \theta)$.
(ii) *Instance $N+1$ is the base network* (G, θ). Exercise 18.6 gives $P(\gamma_{N+1} \mid \theta) = Pr_{\theta}(\gamma)$, and $\gamma \mapsto \gamma_{N+1}$ is a renaming, so $(\alpha \wedge \beta)_{N+1} = \alpha_{N+1} \wedge \beta_{N+1}$ and

$$P(\alpha_{N+1} \mid \beta_{N+1}, \theta) = \frac{Pr_{\theta}(\alpha \wedge \beta)}{Pr_{\theta}(\beta)} = Pr_{\theta}(\alpha \mid \beta),$$

the division legitimate since $P(\beta_{N+1} \mid \theta) > 0$ by assumption.

Problem 18.9. Consider Theorem 18.4. Show that

$$P(D \mid \theta_{X|u}) \;\propto\; \prod_{x} \left(\theta_{x|u}\right)^{D\#(xu)}.$$

Here $\theta_{X|u}$ is a discrete parameter set of the base structure, D is a complete data set of size N, and $D\#(xu)$ is the number of cases in D that contain the family instantiation xu (Definition 17.1). Theorem 18.4 states that

$$P(\theta_{X|u} \mid D) = \eta\, P(\theta_{X|u}) \prod_{x} \left(\theta_{x|u}\right)^{D\#(xu)},$$

where η is a normalizing constant. The proportionality to be shown is with respect to $\theta_{X|u}$: the omitted constant may depend on D but not on the value of the parameter set.

Solution. Read Theorem 18.4 backwards through Bayes rule. Fix $\theta_{X|u}$ with $P(\theta_{X|u}) > 0$ (a zero prior leaves $P(D \mid \theta_{X|u})$ undefined); equating Bayes rule with Theorem 18.4 and cancelling the nonzero prior,

$$P(D \mid \theta_{X|u}) = \eta\, P(D) \prod_{x} \left(\theta_{x|u}\right)^{D\#(xu)},$$

and both η and $P(D)$ are fixed by D alone.
Method (2): directly, exhibiting the constant. By the pruning argument in the proof of Theorem 18.4 (D complete, so every instance node is instantiated), what remains around $\theta_{X|u}$ is an isolated naive Bayes structure with root $\theta_{X|u}$ and children X_i, $i \in I = \{i : u \sim d_i\}$. Split $D = D_1 D_2$ with D_1 the instantiation of those X_i; the children are independent given the root and each has CPT $\theta_{X|u}$ under u, so collecting the $D\#(xu)$ indices whose value is x,

$$P(D_1 \mid \theta_{X|u}) = \prod_{i \in I} \theta_{x_i|u} = \prod_x \left(\theta_{x|u}\right)^{D\#(xu)}.$$

The same pruning gives $P(\theta_{X|u} \mid D) = P(\theta_{X|u} \mid D_1)$, equivalently $P(D_2 \mid D_1, \theta_{X|u}) = P(D_2 \mid D_1)$, whence $P(D \mid \theta_{X|u}) = P(D_2 \mid D_1) \prod_x (\theta_{x|u})^{D\#(xu)}$ with $P(D_2 \mid D_1) = P(D)/P(D_1)$ free of $\theta_{X|u}$.

Problem 18.10. Prove Equation 18.9. That is, let $D = d_1, \ldots, d_N$ be a (possibly incomplete) data set and let θ^k be a parametrization of the base structure G. Definition 18.4 defines the expected count of an event α over base-network variables as

$$D\#(\alpha \mid \theta^k) \stackrel{\text{def}}{=} \sum_{D^c} \Big[[DD^c]\#(\alpha)\Big] P(D^c \mid D, \theta^k),$$

where D^c ranges over the completions of D, each assigning values to exactly those variables that have missing values in D, so that DD^c is a complete data set, and $[DD^c]\#(\alpha)$ is the count of α in that complete data set in the sense of Definition 17.1. Show that

$$D\#(\alpha \mid \theta^k) = \sum_{i=1}^{N} Pr_{\theta^k}(\alpha \mid d_i). \tag{18.9}$$

Solution. Unfold the count into indicators; the exchanged sum is a posterior. Write α_i for α with each base variable X replaced by X_i, and d_i^c for the part of D^c filling case i. Since DD^c has the same N cases, the i-th being the complete instantiation $d_i d_i^c$, Definition 17.1 gives $[DD^c]\#(\alpha) = \sum_i \mathbf{1}[\, d_i d_i^c \models \alpha\,]$, so exchanging the two finite sums,

$$D\#(\alpha \mid \theta^k) = \sum_{i=1}^{N} \sum_{D^c} \mathbf{1}\big[\, d_i d_i^c \models \alpha \,\big]\, P(D^c \mid D, \theta^k) = \sum_{i=1}^{N} P(\alpha_i \mid D, \theta^k),$$

the inner sum collapsing because the completions are mutually exclusive and exhaustive and each DD^c decides α_i. Two reductions finish it.
(i) With Δ_i the instance variables of case i and Γ_i those of the cases $j \neq i$, Exercise 18.5 gives $d\text{-}sep(\Delta_i, \Theta, \Gamma_i)$ – the edges $\theta_{X|u} \to X_j$ are the only ones joining instances, each a diverging valve at a root – so by Theorem 4.2 and $D = d_i D_{-i}$, $P(\alpha_i \mid D, \theta^k) = P(\alpha_i \mid d_i, \theta^k)$.
(ii) Exercise 18.6 gives $P(\gamma_i \mid \theta^k) = Pr_{\theta^k}(\gamma)$; applying it to $\alpha \wedge d_i$ and to d_i and dividing,

$$P(\alpha_i \mid d_i, \theta^k) = \frac{Pr_{\theta^k}(\alpha \wedge d_i)}{Pr_{\theta^k}(d_i)} = Pr_{\theta^k}(\alpha \mid d_i),$$

the denominator positive because Definition 18.4 presupposes $P(D \mid \theta^k) > 0$, which Equation 18.7 factors as $\prod_j Pr_{\theta^k}(d_j)$.

Problem 18.11. Prove Equation 18.27, which appears on Page 514. That is, consider a network structure with families XU, continuous parameter sets with prior Dirichlet densities of exponents $\psi_{x|u}$ and equivalent sample sizes $\psi_{X|u} = \sum_x \psi_{x|u}$, and a complete data set $D = d_1, \ldots, d_N$. Write $D_i = d_1, \ldots, d_i$ (so D_0 is empty and $D_N = D$). The proof of Theorem 18.12 uses Equations 18.14

and 18.16 with the chain rule of Bayesian networks to obtain

$$P(d_i \mid D_{i-1}) = \prod_{xu \sim d_i} \frac{\psi_{x|u} + D_{i-1}\#(xu)}{\psi_{X|u} + D_{i-1}\#(u)},$$

and then the chain rule of probability calculus to obtain

$$P(D) = \prod_{i=1}^{N} P(d_i \mid D_{i-1}) = \prod_{i=1}^{N} \prod_{xu \sim d_i} \frac{\psi_{x|u} + D_{i-1}\#(xu)}{\psi_{X|u} + D_{i-1}\#(u)}.$$

Show that rearranging the terms of this expression gives

$$P(D) = \prod_{XU} \prod_{u} \frac{\prod_x (\psi_{x|u})(\psi_{x|u}+1)\cdots(\psi_{x|u}+D\#(xu)-1)}{(\psi_{X|u})(\psi_{X|u}+1)\cdots(\psi_{X|u}+D\#(u)-1)}. \tag{18.27}$$

Here $xu \sim d_i$ means that the family instantiation xu is compatible with case d_i, and an empty product (when a count is zero) equals 1.

Solution. Pure reindexing: all factors are positive reals and the products finite, so they may be reordered freely. Because D is complete, each family XU has exactly one instantiation $x_i u_i$ compatible with d_i, and the parent instantiations u partition the cases, so

$$P(D) = \prod_{XU} \prod_{u} \prod_{i\,:\,u \sim d_i} \frac{\psi_{x_i|u} + D_{i-1}\#(x_i u)}{\psi_{X|u} + D_{i-1}\#(u)}.$$

Fix XU and u.
(i) *Denominators.* Let $k_1 < \cdots < k_n$ enumerate the indices with $u \sim d_i$, so $n = D\#(u)$; the cases among $d_1, \ldots, d_{k_j - 1}$ compatible with u are exactly $k_1, \ldots, k_{j-1}$, whence $D_{k_j-1}\#(u) = j - 1$ and

$$\prod_{i\,:\,u \sim d_i} \big(\psi_{X|u} + D_{i-1}\#(u)\big) = \prod_{j=0}^{D\#(u)-1} \big(\psi_{X|u} + j\big).$$

(ii) *Numerators.* Partition those cases further by the value x_i of X; for fixed x the same argument gives $D_{i_\ell - 1}\#(xu) = \ell - 1$ along the indices $i_1 < \cdots < i_{D\#(xu)}$ with $xu \sim d_i$, so

$$\prod_{i\,:\,u \sim d_i} \big(\psi_{x_i|u} + D_{i-1}\#(x_i u)\big) = \prod_{x} \prod_{\ell=0}^{D\#(xu)-1} \big(\psi_{x|u} + \ell\big).$$

Both leaves use the empty-product convention when the count is 0 (Check!), and dividing (ii) by (i) then multiplying over u and XU gives (18.27). Since $\Gamma(a+k)/\Gamma(a) = a(a+1)\cdots(a+k-1)$ for $a > 0$ and integer $k \geq 0$ (induction from $\Gamma(a+1) = a\Gamma(a)$), this is Equation 18.21.

Problem 18.12. Let D be an incomplete data set and let D^c denote a completion of D, that is, an instantiation of exactly those variables that have missing values in D, so that DD^c is a complete data set. Show that the expected value of a network parameter $\theta_{x|u}$ given data set D is

$$Ex(\theta_{x|u}) = \sum_{D^c} f(DD^c)\, P(D^c \mid D),$$

where $f(DD^c)$ is the value of the Bayesian estimate $\theta^{be}_{x|u}$ given the complete data set DD^c.

Solution. Case analysis of the posterior on the completions does it, the completions being mutually exclusive and exhaustive since they instantiate exactly the missing variables.
(i) *Discrete parameter sets.* Definition 18.3 makes $Ex(\theta_{x|u}) = \sum_{\theta_{X|u}} \theta_{x|u}\, P(\theta_{X|u} \mid D)$; with $P(\theta_{X|u} \mid D) = \sum_{D^c} P(\theta_{X|u} \mid DD^c) P(D^c \mid D)$ (Section 18.3.3) and the two finite sums exchanged,

$$Ex(\theta_{x|u}) = \sum_{D^c} P(D^c \mid D) \sum_{\theta_{X|u}} \theta_{x|u}\, P(\theta_{X|u} \mid DD^c),$$

whose inner sum is Definition 18.3 for DD^c, that is $f(DD^c)$.
(ii) *Continuous parameter sets.* Definition 18.6 makes $Ex(\theta_{x|u}) = \int \theta_{x|u}\, \rho(\theta_{X|u} \mid D)\, d\theta_{X|u}$, and Section 18.4.6 gives the same case analysis for densities, $\rho(\theta_{X|u} \mid D) = \sum_{D^c} \rho(\theta_{X|u} \mid DD^c) P(D^c \mid D)$; the sum is finite, so it exchanges with the integral and

$$Ex(\theta_{x|u}) = \sum_{D^c} P(D^c \mid D) \int \theta_{x|u}\, \rho(\theta_{X|u} \mid DD^c)\, d\theta_{X|u} = \sum_{D^c} f(DD^c)\, P(D^c \mid D),$$

the inner integral being Definition 18.6 for DD^c.

Problem 18.13. Consider a network structure $A \to B$ where variables A and B are binary, with values a_1, a_2 and b_1, b_2 respectively, and the Dirichlet priors are given by the following exponents:

ψ_{a_1}	ψ_{a_2}	$\psi_{b_1\|a_1}$	$\psi_{b_2\|a_1}$	$\psi_{b_1\|a_2}$	$\psi_{b_2\|a_2}$
10	30	2	2	10	40

Suppose that we are given the following data set D:

	A	B
d_1	a_1	b_2
d_2	a_1	b_1
d_3	a_2	b_1

Compute:
- (a) The Bayesian parameter estimates $\theta^{be}_{a_1}$, $\theta^{be}_{b_1|a_1}$, and $\theta^{be}_{b_1|a_2}$.
- (b) The marginal likelihood $P(D)$.

Solution. $\theta^{be}_{a_1} = 12/43$, $\theta^{be}_{b_1|a_1} = 1/2$, $\theta^{be}_{b_1|a_2} = 11/51$, and $P(D) = 11/5740$. The data set is complete, so part (a) is Equation 18.16 and part (b) is Theorem 18.12.
Counts. The families are A (trivial parent instantiation $\top$) and $B \mid A$ (parent instantiations a_1, a_2):

event	count
$\top$	3
a_1	2
a_2	1
b_1a_1	1
b_2a_1	1
b_1a_2	1
b_2a_2	0

with equivalent sample sizes $\psi_A = 10 + 30 = 40$, $\psi_{B|a_1} = 4$ and $\psi_{B|a_2} = 50$.
(a) Equation 18.16, $\theta^{be}_{x|u} = (\psi_{x|u} + D\#(xu))/(\psi_{X|u} + D\#(u))$, gives

$$\theta^{be}_{a_1} = \frac{10+2}{40+3} = \frac{12}{43} \approx 0.2791,$$
$$\theta^{be}_{b_1|a_1} = \frac{2+1}{4+2} = \frac{1}{2},$$
$$\theta^{be}_{b_1|a_2} = \frac{10+1}{50+1} = \frac{11}{51} \approx 0.2157.$$

(b) Theorem 18.12 contributes one factor per family and parent instantiation, each Gamma ratio being

the rising product $\Gamma(a+k)/\Gamma(a) = a(a+1)\cdots(a+k-1)$ of Exercise 18.11:

$$A: \frac{\Gamma(40)}{\Gamma(43)}\frac{\Gamma(12)}{\Gamma(10)}\frac{\Gamma(31)}{\Gamma(30)} = \frac{(10)(11)(30)}{(40)(41)(42)} = \frac{55}{1148},$$
$$B \mid a_1: \frac{\Gamma(4)}{\Gamma(6)}\frac{\Gamma(3)}{\Gamma(2)}\frac{\Gamma(3)}{\Gamma(2)} = \frac{(2)(2)}{(4)(5)} = \frac{1}{5},$$
$$B \mid a_2: \frac{\Gamma(50)}{\Gamma(51)}\frac{\Gamma(11)}{\Gamma(10)}\frac{\Gamma(40)}{\Gamma(40)} = \frac{10}{50} = \frac{1}{5},$$

the last ratio the empty product 1 because $D\#(b_2a_2) = 0$. Multiplying, $P(D) = \frac{55}{1148}\cdot\frac{1}{5}\cdot\frac{1}{5} = \frac{11}{5740} \approx 0.0019164$.
Method (2) for (b): the chain rule with Theorem 18.8 (Equation 18.14), each factor $P(d_i \mid D_{i-1}) = Pr_{\theta_i^{be}}(d_i)$ evaluated from the earlier cases by Equation 18.16 – $P(d_1) = \frac{10}{40}\cdot\frac{2}{4} = \frac{1}{8}$, $P(d_2 \mid d_1) = \frac{11}{41}\cdot\frac{2}{5} = \frac{22}{205}$, $P(d_3 \mid d_1d_2) = \frac{30}{42}\cdot\frac{10}{50} = \frac{1}{7}$ – whose product is again 11/5740.

Problem 18.14. Consider the network structure $A \to B$ and the Dirichlet priors from Exercise 18.13. That is, A and B are binary with values a_1, a_2 and b_1, b_2, the only edge is $A \to B$, and the prior exponents are

ψ_{a_1}	ψ_{a_2}	$\psi_{b_1\mid a_1}$	$\psi_{b_2\mid a_1}$	$\psi_{b_1\mid a_2}$	$\psi_{b_2\mid a_2}$
10	30	2	2	10	40

Compute the expected values of parameters θ_{a_1}, $\theta_{b_1\mid a_1}$, and $\theta_{b_1\mid a_2}$ given the following data set:

	A	B
d_1	a_1	b_2
d_2	?	b_1
d_3	a_2	?

Hint: perform case analysis on the missing values in the data set.

Solution. $Ex(\theta_{a_1}) = 605/2279$, $Ex(\theta_{b_1\mid a_1}) = 117/265$, and $Ex(\theta_{b_1\mid a_2}) = 2827/13515$. By Exercise 18.12,

$$Ex(\theta_{x\mid u}) = \sum_{D^c} f(DD^c)\,P(D^c \mid D), \qquad P(D^c \mid D) = \frac{P(DD^c)}{\sum_{D^c} P(DD^c)},$$

with f given by Equation 18.16 and each $P(DD^c)$ a complete-data marginal likelihood from Theorem 18.12, as in Exercise 18.13; the denominator is $P(D)$ because the four completions are mutually exclusive and exhaustive.
The four completed data sets and their counts.

comp.	A_2	B_3	$\#a_1$	$\#a_2$	$\#b_1a_1$	$\#b_2a_1$	$\#b_1a_2$	$\#b_2a_2$
D_1^c	a_1	b_1	2	1	1	1	1	0
D_2^c	a_1	b_2	2	1	1	1	0	1
D_3^c	a_2	b_1	1	2	0	1	2	0
D_4^c	a_2	b_2	1	2	0	1	1	1

Marginal likelihoods. With $\psi_A = 40$, $\psi_{B\mid a_1} = 4$, $\psi_{B\mid a_2} = 50$ and each Gamma ratio a rising product (Exercise 18.11), the family A factor depends only on $(\#a_1, \#a_2)$:

$$\frac{(10)(11)\,(30)}{(40)(41)(42)} = \frac{55}{1148} \text{ at } (2,1), \qquad \frac{(10)\,(30)(31)}{(40)(41)(42)} = \frac{155}{1148} \text{ at } (1,2).$$

The $B \mid a_1$ factor is $\frac{(2)(2)}{(4)(5)} = \frac{1}{5}$ when $\#a_1 = 2$ with one case of each value of B, and $\frac{2}{4} = \frac{1}{2}$ when $\#a_1 = 1$ with that case b_2; the $B \mid a_2$ factor is

$$\frac{10}{50} = \frac{1}{5}, \quad \frac{40}{50} = \frac{4}{5}, \quad \frac{(10)(11)}{(50)(51)} = \frac{11}{255}, \quad \frac{(10)(40)}{(50)(51)} = \frac{8}{51}$$

for $D_1^c, \ldots, D_4^c$ respectively. Multiplying,

completion	A factor	$B \mid a_1$	$B \mid a_2$	$P(DD^c)$	value
D_1^c	55/1148	1/5	1/5	11/5740	0.0019164
D_2^c	55/1148	1/5	4/5	11/1435	0.0076655
D_3^c	155/1148	1/2	11/255	341/117096	0.0029121
D_4^c	155/1148	1/2	8/51	155/14637	0.0105896

Summing gives the marginal likelihood of the incomplete data set,

$$P(D) = \frac{11}{5740} + \frac{11}{1435} + \frac{341}{117096} + \frac{155}{14637} = \frac{53}{2296} \approx 0.0230836,$$

and dividing each row by it gives the posterior over completions. Put over the common denominator $13515 = 5 \cdot 51 \cdot 53$:

$$\begin{aligned} P(D_1^c \mid D) &= \tfrac{1122}{13515} = \tfrac{22}{265} \approx 0.08302, \\ P(D_2^c \mid D) &= \tfrac{4488}{13515} = \tfrac{88}{265} \approx 0.33208, \\ P(D_3^c \mid D) &= \tfrac{1705}{13515} = \tfrac{341}{2703} \approx 0.12616, \\ P(D_4^c \mid D) &= \tfrac{6200}{13515} = \tfrac{1240}{2703} \approx 0.45875, \end{aligned}$$

Complete-data estimates. By Equation 18.16, with $D\#(\top) = 3$ throughout:

comp.	θ_{a_1}	$\theta_{b_1\mid a_1}$	$\theta_{b_1\mid a_2}$
D_1^c	12/43	$3/6 = 1/2$	11/51
D_2^c	12/43	$3/6 = 1/2$	10/51
D_3^c	11/43	2/5	$12/52 = 3/13$
D_4^c	11/43	2/5	11/52

The entries are $(10 + \#a_1)/43$, $(2 + \#b_1a_1)/(4 + \#a_1)$, and $(10 + \#b_1a_2)/(50 + \#a_2)$ with the counts from the previous table (Check!).

The expectations. Averaging each column against the posterior over completions:

$$\begin{aligned} Ex(\theta_{a_1}) &= \frac{12}{43} \cdot \frac{1122 + 4488}{13515} + \frac{11}{43} \cdot \frac{1705 + 6200}{13515} \\ &= \frac{12 \cdot 5610 + 11 \cdot 7905}{43 \cdot 13515} = \frac{154275}{581145} = \frac{605}{2279} \approx 0.26547. \end{aligned}$$

$$\begin{aligned} Ex(\theta_{b_1\mid a_1}) &= \frac{1}{2} \cdot \frac{5610}{13515} + \frac{2}{5} \cdot \frac{7905}{13515} \\ &= \frac{2805 + 3162}{13515} = \frac{5967}{13515} = \frac{117}{265} \approx 0.44151. \end{aligned}$$

$$\begin{aligned} Ex(\theta_{b_1\mid a_2}) &= \frac{11}{51} \cdot \frac{1122}{13515} + \frac{10}{51} \cdot \frac{4488}{13515} \\ &\quad + \frac{3}{13} \cdot \frac{1705}{13515} + \frac{11}{52} \cdot \frac{6200}{13515} \\ &= \frac{1}{13515}\left(242 + 880 + \frac{5115}{13} + \frac{17050}{13}\right) \\ &= \frac{1}{13515}\left(242 + 880 + \frac{22165}{13}\right) \\ &= \frac{242 + 880 + 1705}{13515} = \frac{2827}{13515} \approx 0.20917. \end{aligned}$$

17.3 Exercises 18.15–18.21

Problem 18.15. Consider a meta-network that induces a meta-distribution P, let D be an incomplete data set, and let θ be a parametrization. Describe a Gibbs sampler that can be used for estimating the expectation of $P(\theta \mid D, D_c)$ with respect to the distribution $P(D_c \mid D)$, where D_c is a completion of the data set D. That is, describe a Gibbs-Markov chain for the distribution $P(C \mid D)$, where C are the variables with missing values in D, and then show how we can simulate this chain.

Solve this problem for both discrete and continuous parameter sets.
Recall the setting of Definition 18.2: the meta-network for a base structure G with families XU contains a root node $\theta_{X|u}$ for every family XU and every parent instantiation u, together with N copies (slices) of the base network, the i-th slice containing a node X_i for every base variable X. The parents of X_i are U_i together with the parameter sets $\theta_{X|u}$, and $P(x_i \mid u_i, \theta_{X|u_i}) = \theta_{x|u_i}$. A case d_i of D is a partial instantiation of the slice-i variables, and a completion D_c assigns values to exactly the variables that are missing in D.

Solution. Run a collapsed Gibbs chain whose state is C alone, with θ marginalized out. An instantiation c of $C = C^1, \ldots, C^m$, the variables missing in D, is exactly a completion D_c, so $P(C \mid D)$ and $P(D_c \mid D)$ are one distribution and the target $P(\theta \mid D) = \sum_{D_c} P(\theta \mid D, D_c) P(D_c \mid D)$ is an expectation under it.
The chain. One sweep visits each C^j in turn and redraws it from $P(C^j \mid D, c^{-j})$, $c^{-j} = c \setminus c^j$ – the Gibbs construction of Chapter 15 applied to the meta-network under evidence D. Each single-variable update is reversible with respect to $P(C \mid D)$, so the sweep leaves $P(C \mid D)$ invariant; the chain is ergodic once every such conditional is positive, automatic under Dirichlet priors since $\psi_{x|u} \geq 1$ makes the density positive on the interior of the simplex.
Simulating it. Since C^j is a single base variable, $P(C^j = z \mid D, c^{-j}) = \eta\, P(D\, c^{-j}\, z)$ with η normalizing over the values z and $D\, c^{-j} z$ complete. So only complete-data marginal likelihoods are needed, and for complete data the parameter sets decouple (Theorem 18.1, by the pruning of Figure 18.3), giving:

- discrete parameter sets, from $P(\theta) = \prod_{XU} \prod_u P(\theta_{X|u})$ (18.5) and $P(D_c \mid \theta) = \prod_{xu} \theta_{x|u}^{D_c \#(xu)}$ (Exercise 18.9),
$$P(D_c) \;=\; \prod_{XU} \prod_u \Big(\sum_{\theta_{X|u}} P(\theta_{X|u}) \prod_x \theta_{x|u}^{D_c \#(xu)} \Big);$$
- continuous parameter sets with Dirichlet priors (Definition 18.5), the closed form (18.21),
$$P(D_c) \;=\; \prod_{XU} \prod_u \frac{\Gamma(\psi_{X|u})}{\Gamma\big(\psi_{X|u} + D_c \#(u)\big)} \prod_x \frac{\Gamma\big(\psi_{x|u} + D_c \#(xu)\big)}{\Gamma(\psi_{x|u})}.$$

Only ratios across z are needed, and flipping one missing X_i moves only the counts of the family XU and of the families of the children of X, by ± 1, so with $\Gamma(z{+}1) = z\Gamma(z)$ everything else cancels: writing $\hat{D}$ for the complete data set with X_i removed from all counts and $v_i(z)$ for the parent instantiation of Y_i when $X_i = z$,

$$P\big(X_i = z \mid D, c^{-j}\big) \;=\; \eta \cdot \frac{\psi_{z|u_i} + \hat{D}\#(z u_i)}{\psi_{X|u_i} + \hat{D}\#(u_i)} \prod_{Y \in \mathrm{ch}(X)} \frac{\psi_{y_i|v_i(z)} + \hat{D}\#\big(y_i\, v_i(z)\big)}{\psi_{Y|v_i(z)} + \hat{D}\#\big(v_i(z)\big)},$$

at cost $O(|X| \cdot |\mathrm{ch}(X)|)$ per variable.
The estimator. Discard a burn-in prefix, collect $c^{(1)}, \ldots, c^{(T)}$, and take $P(\theta \mid D) \approx \frac{1}{T} \sum_t P(\theta \mid D, c^{(t)})$, each term closed-form because $D\, c^{(t)}$ is complete: Theorems 18.1 and 18.4 give

$$P\big(\theta \mid D\, c^{(t)}\big) = \prod_{XU} \prod_u \eta\, P(\theta_{X|u}) \prod_x \theta_{x|u}^{D\, c^{(t)} \#(xu)}$$

in the discrete case, and Theorem 18.9 makes each continuous factor Dirichlet with exponents $\psi_{x|u} + D\, c^{(t)} \#(xu)$:

$$\rho(\theta \mid D) \;\approx\; \frac{1}{T} \sum_{t=1}^{T} \prod_{XU} \prod_u \rho_{\mathrm{Dir}}\Big(\theta_{X|u}\,;\, \big\{\psi_{x|u} + D\, c^{(t)} \#(xu)\big\}_x\Big).$$

Problem 18.16. Identify the relationship between the expected counts of Definition 18.4 and the expected empirical distribution of Definition 17.2.
For reference, Definition 18.4 defines the expected count of an event α given parameter estimates θ^k

as

$$D\#(\alpha \mid \theta^k) \stackrel{\text{def}}{=} \sum_{D_c} \Big[[DD_c]\#(\alpha) \Big] P(D_c \mid D, \theta^k),$$

where D_c ranges over the completions of D and $[DD_c]\#(\alpha)$ is the ordinary count of Definition 17.1 in the completed data set. Definition 17.2 defines the expected empirical distribution of D under θ^k as

$$\Pr_{D,\theta^k}(\alpha) \stackrel{\text{def}}{=} \frac{1}{N} \sum_{d_i, c_i \models \alpha} \Pr_{\theta^k}(c_i \mid d_i),$$

where C_i are the variables with missing values in case d_i and the sum ranges over pairs (i, c_i) with c_i an instantiation of C_i such that $d_i c_i \models \alpha$.

Solution. The relationship is

$$D\#(\alpha \mid \theta^k) \;=\; N \cdot \Pr_{D,\theta^k}(\alpha),$$

exactly as $D\#(\alpha) = N \cdot \Pr_D(\alpha)$ in Definition 17.1.
A completion D_c is a tuple $(c_1, \ldots, c_N)$ with c_i an instantiation of C_i. By Exercise 18.5 the parameter-set roots d-separate the slices of the meta-network from one another, and by Exercise 18.6 each slice given θ^k is the base network (G, θ^k), so the N cases are independent given θ^k: $P(D_c \mid D, \theta^k) = \prod_j \Pr_{\theta^k}(c_j \mid d_j)$. With $[DD_c]\#(\alpha) = \sum_i [\, d_i c_i \models \alpha \,]$ in Iverson brackets (Definition 17.1), Definition 18.4 and an exchange of summations give

$$D\#(\alpha \mid \theta^k) = \sum_{i=1}^{N} \sum_{c_1, \ldots, c_N} [\, d_i c_i \models \alpha \,] \prod_{j=1}^{N} \Pr_{\theta^k}(c_j \mid d_j).$$

In the i-th inner sum the c_j, $j \neq i$, enter only through the factors $\Pr_{\theta^k}(c_j \mid d_j)$, which sum to 1, so they marginalize away and

$$D\#(\alpha \mid \theta^k) = \sum_{i=1}^{N} \sum_{c_i \,:\, d_i c_i \models \alpha} \Pr_{\theta^k}(c_i \mid d_i) = \sum_{d_i, c_i \models \alpha} \Pr_{\theta^k}(c_i \mid d_i) = N \cdot \Pr_{D,\theta^k}(\alpha),$$

the last step being precisely Definition 17.2.

Problem 18.17. Show that Algorithm 50, ML EM, is a special case of Algorithm 52, MAP EM C, in the case where all parameter sets $\theta_{X|u}$ have noninformative priors (i.e., all Dirichlet exponents are equal to 1). Note: in this case, the fixed points of Algorithm 52 are also stationary points of the log-likelihood function.
For reference, the two algorithms differ only in their accumulation and update lines. Both take a structure G with families XU, initial estimates θ^0, and a data set D of size N, and both iterate the following until convergence. Algorithm 50 (ML EM) sets $c_{xu} \leftarrow 0$ for every family instantiation xu, accumulates $c_{xu} \leftarrow c_{xu} + \Pr_{\theta^k}(xu \mid d_i)$ over the cases d_i, and then updates

$$\theta^{k+1}_{x|u} \;=\; \frac{c_{xu}}{\sum_{x^\star} c_{x^\star u}}.$$

Algorithm 52 (MAP EM C) takes in addition a Dirichlet prior $\psi_{x|u}/\psi_{X|u}$ for each parameter set, sets $c_{xu} \leftarrow 0$ and $c_u \leftarrow 0$, accumulates both $c_{xu} \leftarrow c_{xu} + \Pr_{\theta^k}(xu \mid d_i)$ and $c_u \leftarrow c_u + \Pr_{\theta^k}(u \mid d_i)$ over the cases, and then updates

$$\theta^{k+1}_{x|u} \;=\; \frac{c_{xu} + \psi_{x|u} - 1}{c_u + \psi_{X|u} - |X|},$$

where $\psi_{X|u} = \sum_x \psi_{x|u}$ and $|X|$ is the number of values of X.

Solution. With $\psi_{x|u} \equiv 1$ the two algorithms agree line for line, hence generate the same sequence $\theta^0, \theta^1, \ldots$ from the same θ^0.
Accumulators. Both compute $c_{xu} = \sum_i \Pr_{\theta^k}(xu \mid d_i)$, which by (18.9) is $D\#(xu \mid \theta^k)$. Algorithm 52's extra $c_u = D\#(u \mid \theta^k)$ is redundant: for fixed u the events $x^\star u$ are mutually exclusive with disjunction u, so

$$c_u \;=\; \sum_{i=1}^{N} \Pr_{\theta^k}(u \mid d_i) \;=\; \sum_{i=1}^{N} \sum_{x^\star} \Pr_{\theta^k}(x^\star u \mid d_i) \;=\; \sum_{x^\star} c_{x^\star u},$$

Algorithm 50's denominator.
Updates. All exponents 1 give $\psi_{x|u} - 1 = 0$ and $\psi_{X|u} - |X| = 0$, so Algorithm 52's update collapses to $\theta^{k+1}_{x|u} = c_{xu}/c_u = c_{xu}/\sum_{x^\star} c_{x^\star u}$, which is Algorithm 50's. The remaining lines – initialization, the case loop, the convergence test, the return – are identical.
The note. By Definition 18.5 exponents all 1 make each $\rho(\theta_{X|u})$ the constant density on the simplex, so $\rho(\theta)$ is constant by (18.18) and $\rho(\theta \mid D) = \eta'\, P(D \mid \theta)$; hence $\log \rho(\theta \mid D)$ and $LL(\theta \mid D) = \log P(D \mid \theta)$ differ by an additive constant and share their stationary points on the feasible set.

Problem 18.18. The Hessian matrix of the Laplace approximation is specified in terms of three types of second partial derivatives, with respect to:

- a single parameter $\theta_{x|u}$;
- parameters $\theta_{x|u}$ and $\theta_{x^\star|u}$ of the same parameter set $\theta_{X|u}$, where $x \neq x^\star$;
- parameters $\theta_{x|u}$ and $\theta_{y|v}$ of two different parameter sets $\theta_{X|u}$ and $\theta_{Y|v}$.

(a) Show that the partial derivative of the Dirichlet density of parameter set $\theta_{X|u}$ with respect to parameter $\theta_{x|u}$ is

$$\frac{\partial \rho(\theta_{X|u})}{\partial \theta_{x|u}} \;=\; \frac{\psi_{x|u} - 1}{\theta_{x|u}} \cdot \rho(\theta_{X|u}).$$

(b) Given (a), show that the partial derivative of the prior density of network parameters $\rho(\theta)$ is

$$\frac{\partial \rho(\theta)}{\partial \theta_{x|u}} \;=\; \frac{\psi_{x|u} - 1}{\theta_{x|u}} \cdot \rho(\theta).$$

(c) Given (a) and (b), show how to compute the second partial derivatives

$$\frac{\partial^2 \rho(\theta)}{\partial \theta_{x|u}\, \partial \theta_{x|u}}, \qquad \frac{\partial^2 \rho(\theta)}{\partial \theta_{x|u}\, \partial \theta_{x^\star|u}}, \qquad \frac{\partial^2 \rho(\theta)}{\partial \theta_{x|u}\, \partial \theta_{y|v}}.$$

Here $\rho(\theta_{X|u}) = \eta \prod_x \theta_{x|u}^{\psi_{x|u}-1}$ is the Dirichlet density of Definition 18.5 with exponents $\psi_{x|u} \geq 1$, and $\rho(\theta) = \prod_{XU} \prod_u \rho(\theta_{X|u})$ is the prior density of network parameters, as given by (18.18).

Solution. Abbreviate $a_{x|u} = \psi_{x|u} - 1 \geq 0$.
(a) Only the factor indexed by x of $\rho(\theta_{X|u}) = \eta \prod_{x^\star} \theta_{x^\star|u}^{a_{x^\star|u}}$ (Definition 18.5) depends on $\theta_{x|u}$, so

$$\frac{\partial \rho(\theta_{X|u})}{\partial \theta_{x|u}} = \eta\, a_{x|u}\, \theta_{x|u}^{a_{x|u}-1} \prod_{x^\star \neq x} \theta_{x^\star|u}^{a_{x^\star|u}} = \frac{\psi_{x|u} - 1}{\theta_{x|u}} \cdot \rho(\theta_{X|u}),$$

multiplying and dividing by $\theta_{x|u} > 0$ on the interior.
(b) Distinct parameter sets involve disjoint parameters, so exactly one factor of $\rho(\theta) = \prod_{YV} \prod_v \rho(\theta_{Y|v})$ (18.18) depends on $\theta_{x|u}$; with R the product of the others (free of $\theta_{x|u}$) and (a),

$$\frac{\partial \rho(\theta)}{\partial \theta_{x|u}} = R \cdot \frac{a_{x|u}}{\theta_{x|u}} \rho(\theta_{X|u}) = \frac{\psi_{x|u} - 1}{\theta_{x|u}} \cdot \rho(\theta).$$

(c) Differentiate (b) once more; all that matters is whether $1/\theta_{x|u}$ depends on the second differentiation variable.

(i) *A single parameter $\theta_{x|u}$.* It does, so the product rule gives two terms:

$$\begin{aligned}\frac{\partial^2\rho(\theta)}{\partial\theta_{x|u}\partial\theta_{x|u}} &= -\frac{a_{x|u}}{\theta^2_{x|u}}\rho(\theta) + \frac{a_{x|u}}{\theta_{x|u}}\cdot\frac{a_{x|u}}{\theta_{x|u}}\rho(\theta)\\ &= \frac{(\psi_{x|u}-1)(\psi_{x|u}-2)}{\theta^2_{x|u}}\cdot\rho(\theta).\end{aligned}$$

(ii), (iii) *Two distinct parameters $\theta_{x|u}, \theta_{y|v}$, whether of the same set ($\theta_{y|v} = \theta_{x^\star|u}$, $x \neq x^\star$) or of different sets.* Now $1/\theta_{x|u}$ is constant in $\theta_{y|v}$, so one term survives and (b) applies again to $\rho(\theta)$:

$$\frac{\partial^2\rho(\theta)}{\partial\theta_{x|u}\partial\theta_{y|v}} = \frac{a_{x|u}}{\theta_{x|u}}\cdot\frac{\partial\rho(\theta)}{\partial\theta_{y|v}} = \frac{(\psi_{x|u}-1)(\psi_{y|v}-1)}{\theta_{x|u}\,\theta_{y|v}}\cdot\rho(\theta).$$

Problem 18.19. Analogously to Exercise 18.18, consider partial derivatives of the likelihood of network parameters $P(D \mid \theta)$.
(a) Show that the partial derivative of the likelihood $P(D \mid \theta)$ with respect to parameter $\theta_{x|u}$ is

$$\frac{\partial P(D \mid \theta)}{\partial\theta_{x|u}} \;=\; P(D \mid \theta)\cdot\frac{D\#(xu \mid \theta)}{\theta_{x|u}}.$$

(b) Given (a), show how to compute the second partial derivatives

$$\frac{\partial^2 P(D \mid \theta)}{\partial\theta_{x|u}\,\partial\theta_{x|u}}, \qquad \frac{\partial^2 P(D \mid \theta)}{\partial\theta_{x|u}\,\partial\theta_{x^\star|u}}, \qquad \frac{\partial^2 P(D \mid \theta)}{\partial\theta_{x|u}\,\partial\theta_{y|v}},$$

the second being with respect to two parameters $\theta_{x|u}, \theta_{x^\star|u}$ ($x \neq x^\star$) of the same parameter set, and the third with respect to parameters of two different parameter sets $\theta_{X|u}$ and $\theta_{Y|v}$.
Here $P(D \mid \theta) = \prod_{i=1}^N \Pr_\theta(d_i)$ as given by (18.20) (equivalently (18.7)), D is a data set of size N that need not be complete, and $D\#(\alpha \mid \theta)$ is the expected count of Definition 18.4, computable by (18.9) as $D\#(\alpha \mid \theta) = \sum_{i=1}^N \Pr_\theta(\alpha \mid d_i)$.

Solution. Two facts about the base network's polynomial do all the work, both resting on its multilinearity in the parameters (a term is a complete instantiation, so two parameters of one family never share a term). For any evidence e:

- by Theorem 12.2, equation (12.4), $\frac{\partial \Pr_\theta(e)}{\partial\theta_{x|u}} = \frac{\Pr_\theta(xu,e)}{\theta_{x|u}}$;
- second derivatives in two parameters of the same family vanish (Exercise 12.4(b)), and otherwise (Exercise 12.5) $\frac{\partial^2 \Pr_\theta(e)}{\partial\theta_{x|u}\partial\theta_{y|v}} = \frac{\Pr_\theta(xu,yv,e)}{\theta_{x|u}\,\theta_{y|v}}$.

The second fact's two cases are one statement, since distinct parameters of the same family make xu, yv inconsistent and so $\Pr_\theta(xu, yv, e) = 0$; only $\theta_{x|u} = \theta_{y|v}$ is exceptional. Abbreviate $p_i = \Pr_\theta(d_i)$, so $P(D \mid \theta) = \prod_i p_i$, and put

$$\begin{aligned}D\#(xu \mid \theta) &= \sum_{i=1}^N \Pr_\theta(xu \mid d_i),\\ D\#(xu, yv \mid \theta) &= \sum_{i=1}^N \Pr_\theta(xu \wedge yv \mid d_i),\\ S(xu, yv \mid \theta) &= \sum_{i=1}^N \Pr_\theta(xu \mid d_i)\Pr_\theta(yv \mid d_i),\end{aligned}$$

the first two expected counts (Definition 18.4, evaluated by (18.9)), the third a sum of products of two per-case posteriors rather than a count.

Part (a). Differentiating $\prod_i p_i$ and dividing by it,

$$\frac{\partial P(D \mid \theta)}{\partial \theta_{x|u}} = P(D \mid \theta) \sum_{i=1}^{N} \frac{1}{p_i} \frac{\partial p_i}{\partial \theta_{x|u}} = P(D \mid \theta) \sum_{i=1}^{N} \frac{\Pr_\theta(xu \mid d_i)}{\theta_{x|u}},$$

by the first fact with $e = d_i$, and the sum is $D\#(xu \mid \theta)/\theta_{x|u}$.
Part (b). For any two parameters θ_1, θ_2, differentiating $\prod_i p_i$ twice, dividing by it, and writing $\alpha_i = (\partial p_i/\partial \theta_1)/p_i$, $\beta_i = (\partial p_i/\partial \theta_2)/p_i$, $\gamma_i = (\partial^2 p_i/\partial \theta_1 \partial \theta_2)/p_i$,

$$\frac{\partial^2 P(D \mid \theta)}{\partial \theta_1 \partial \theta_2} = P(D \mid \theta) \left[\Big(\sum_i \alpha_i\Big)\Big(\sum_i \beta_i\Big) - \sum_i \alpha_i \beta_i + \sum_i \gamma_i \right],$$

the correction $-\sum_i \alpha_i \beta_i$ deleting the diagonal $i = j$, which the product of sums contains but the genuine cross terms $i \neq j$ do not. With $\theta_1 = \theta_{x|u}$, $\theta_2 = \theta_{y|v}$ the two facts give

$$\alpha_i = \frac{\Pr_\theta(xu \mid d_i)}{\theta_{x|u}}, \quad \beta_i = \frac{\Pr_\theta(yv \mid d_i)}{\theta_{y|v}}, \quad \gamma_i = \frac{\Pr_\theta(xu \wedge yv \mid d_i)}{\theta_{x|u}\,\theta_{y|v}},$$

with $\gamma_i \equiv 0$ whenever the two parameters belong to the same family (including $\theta_1 = \theta_2$, where p_i is linear in that parameter). Hence:
(i), (ii) *Same set: $\theta_{x|u}$ with itself, or with $\theta_{x^\star|u}$, $x \neq x^\star$.* Here $\gamma_i = 0$, so

$$\frac{\partial^2 P(D \mid \theta)}{\partial \theta_{x|u} \partial \theta_{x^\star|u}} = P(D \mid \theta) \cdot \frac{D\#(xu \mid \theta)\, D\#(x^\star u \mid \theta) - S(xu, x^\star u \mid \theta)}{\theta_{x|u}\,\theta_{x^\star|u}},$$

which at $x^\star = x$ reads $P(D \mid \theta)\big[D\#(xu \mid \theta)^2 - \sum_i \Pr_\theta(xu \mid d_i)^2\big]/\theta_{x|u}^2$.
(iii) *Different sets, $\theta_{x|u}$ and $\theta_{y|v}$.* The joint term survives:

$$\frac{\partial^2 P(D \mid \theta)}{\partial \theta_{x|u} \partial \theta_{y|v}} = P(D \mid \theta) \cdot \frac{D\#(xu \mid \theta) D\#(yv \mid \theta) - S(xu, yv \mid \theta) + D\#(xu, yv \mid \theta)}{\theta_{x|u}\,\theta_{y|v}}.$$

Problem 18.20. Consider the second partial derivative

$$\frac{\partial^2 \log \rho(D, \theta)}{\partial \theta_{x|u}\, \partial \theta_{y|v}}(\theta^{ma})$$

used in the Laplace approximation. Show how this derivative can be computed by performing inference on a Bayesian network with parametrization θ^{ma}. What is the complexity of computing all the second partial derivatives (i.e., computing the Hessian)? Hint: use the fact that $\rho(D, \theta) = P(D \mid \theta)\rho(\theta)$ and consider Exercises 18.18 and 18.19.
Recall from (18.22) that the Laplace approximation is

$$\log P(D) \approx \log P(D \mid \theta^{ma}) + \log \rho(\theta^{ma}) + \frac{d}{2} \log(2\pi) - \frac{1}{2} \log |\Sigma|,$$

where d is the number of independent parameters, θ^{ma} are the MAP estimates, and Σ is the matrix whose entries are the negated second partial derivatives above, evaluated at θ^{ma}.

Solution. Every entry is a sum of per-case family posteriors in the base network (G, θ^{ma}) plus a prior term needing no inference. Since $\log \rho(D, \theta) = \log P(D \mid \theta) + \log \rho(\theta)$, the two halves are Exercises 18.19 and 18.18 on the log scale.
The prior term. By Definition 18.5 and (18.18), $\log \rho(\theta) = \log \eta + \sum_{XU} \sum_u \sum_x (\psi_{x|u} - 1) \log \theta_{x|u}$, in

which each parameter appears in exactly one summand, so

$$\frac{\partial^2 \log \rho(\theta)}{\partial\theta_{x|u}\,\partial\theta_{y|v}} = \begin{cases} -\dfrac{\psi_{x|u}-1}{\theta_{x|u}^2}, & \theta_{y|v}=\theta_{x|u}, \\ 0, & \text{otherwise.} \end{cases}$$

The prior thus touches only the diagonal of Σ, from the exponents and the MAP estimates alone.
The likelihood term. Differentiating twice through $\partial^2 \log f = f^{-1}\partial^2 f - (f^{-1}\partial_1 f)(f^{-1}\partial_2 f)$, the bracket of Exercise 18.19(b) loses its $(\sum_i \alpha_i)(\sum_i \beta_i)$ term:

$$\frac{\partial^2 \log P(D \mid \theta)}{\partial\theta_1\,\partial\theta_2} = \sum_{i=1}^{N} \big(\gamma_i - \alpha_i\beta_i\big),$$

with $\alpha_i, \beta_i, \gamma_i$ the per-case ratios computed there from Theorem 12.2, equation (12.4). For $\theta_1 = \theta_{x|u}$, $\theta_2 = \theta_{y|v}$ distinct this is

$$\frac{\partial^2 \log P(D \mid \theta)}{\partial\theta_{x|u}\,\partial\theta_{y|v}} = \frac{1}{\theta_{x|u}\,\theta_{y|v}} \sum_{i=1}^{N} \Big[\Pr_\theta(xu \wedge yv \mid d_i) - \Pr_\theta(xu \mid d_i)\Pr_\theta(yv \mid d_i)\Big],$$

a sum of per-case covariances of the indicators of xu and yv, whose first term vanishes for two distinct parameters of one set; and for $\theta_1 = \theta_2 = \theta_{x|u}$, where $\gamma_i = 0$ and $\alpha_i = \beta_i$,

$$\frac{\partial^2 \log P(D \mid \theta)}{\partial\theta_{x|u}\,\partial\theta_{x|u}} = -\frac{1}{\theta_{x|u}^2} \sum_{i=1}^{N} \Pr_\theta(xu \mid d_i)^2.$$

Adding the two contributions at θ^{ma} gives the Hessian entries. For $\theta_{y|v} \neq \theta_{x|u}$,

$$\frac{\partial^2 \log \rho(D,\theta)}{\partial\theta_{x|u}\partial\theta_{y|v}}(\theta^{ma}) = \frac{\displaystyle\sum_{i=1}^{N} \Big[\Pr_{\theta^{ma}}(xu \wedge yv \mid d_i) - \Pr_{\theta^{ma}}(xu \mid d_i)\Pr_{\theta^{ma}}(yv \mid d_i)\Big]}{\theta^{ma}_{x|u}\,\theta^{ma}_{y|v}},$$

and on the diagonal,

$$\frac{\partial^2 \log \rho(D,\theta)}{\partial\theta_{x|u}\partial\theta_{x|u}}(\theta^{ma}) = -\frac{1}{\big(\theta^{ma}_{x|u}\big)^2}\left[\sum_{i=1}^{N} \Pr_{\theta^{ma}}(xu \mid d_i)^2 + \psi_{x|u} - 1\right].$$

How inference supplies these. Every quantity is a posterior in the base network (G, θ^{ma}) under evidence d_i; the meta-network is never used.

- $\Pr_{\theta^{ma}}(xu \mid d_i)$ is a family marginal, so one jointree propagation per case (or one circuit evaluation with its derivatives, Theorem 12.2) yields all family marginals for that case, and N propagations give every $\sum_i \Pr(xu \mid d_i)\Pr(yv \mid d_i)$.
- $\Pr_{\theta^{ma}}(xu \wedge yv \mid d_i)$ is a two-family joint marginal: read off directly if both families share a cluster, else by the chain rule $\Pr(xu \wedge yv \mid d_i) = \Pr(xu \mid d_i \wedge yv)\Pr(yv \mid d_i)$, so one extra propagation with evidence $d_i \wedge yv$ delivers the whole Hessian row indexed by $\theta_{y|v}$ for case d_i.

Complexity. With d independent parameters, n variables and width w: Nd propagations at $O(n\exp(w))$ each, one per case and per family instantiation yv used as auxiliary evidence; $O(Nd^2)$ arithmetic to accumulate the $O(d^2)$ entries, each a sum over N cases of two products; and $O(d^3)$ for $|\Sigma|$. The total is $O(Nd\,n\exp(w) + Nd^2 + d^3)$, dominated in practice by $Nd\,n\exp(w)$ – a factor of d more inference than one EM iteration, which is why the text sets these entries to 0 for $X \neq Y$ and works with a block-diagonal Σ.

Problem 18.21. Extend the structural meta-network in Figure 18.6 to allow for reasoning about incomplete data sets. That is, the meta-network needs to have additional nodes to allow us to assert an incomplete data set as evidence.
The structural meta-network of Figure 18.6(b) is built for a problem with two binary variables A and B and has the following structure. There is a root node Ω with three values G_1, G_2, G_3, corresponding to the three possible network structures over A and B, and six root parameter-set nodes $\theta_{A|b_1}, \theta_{A|b_2}, \theta_A, \theta_B, \theta_{B|a_1}, \theta_{B|a_2}$. There is one further node A_iB_i per case $i = 1, \ldots, N$; it is a single lumped variable with the four values $a_1b_1, a_1b_2, a_2b_1, a_2b_2$, and its parents are Ω together with all six parameter sets. The correspondence between structures and parameter sets is

$$\begin{aligned}
\Omega = G_1 &: \text{ structure } A,\ B \qquad \text{with parameter sets } \theta_A,\ \theta_B;\\
\Omega = G_2 &: \text{ structure } A \to B \quad \text{with } \theta_A,\ \theta_{B|a_1},\ \theta_{B|a_2};\\
\Omega = G_3 &: \text{ structure } B \to A \quad \text{with } \theta_B,\ \theta_{A|b_1},\ \theta_{A|b_2}.
\end{aligned}$$

So the CPT of A_iB_i is $P(ab \mid \Omega = G_1, \theta) = \theta_a\theta_b$, $P(ab \mid \Omega = G_2, \theta) = \theta_a\theta_{b|a}$, and $P(ab \mid \Omega = G_3, \theta) = \theta_b\theta_{a|b}$. By contrast, the parameter meta-network of Figure 18.6(a), which assumes the fixed structure $A \to B$, keeps A_i and B_i as separate nodes: θ_A is a parent of every A_i, A_i is a parent of B_i, and $\theta_{B|a_1}, \theta_{B|a_2}$ are parents of every B_i.

Solution. Give each lumped node its components back as deterministic children. In Figure 18.6(b) the lumping of A and B into A_iB_i is unavoidable – which of A_i, B_i is the other's parent depends on Ω, while a Bayesian network has a fixed DAG – so "only A observed" is the disjunction $A_iB_i \in \{a_1b_1, a_1b_2\}$ rather than an instantiation.
The construction. For each case i add two leaves whose single parent is A_iB_i: A_i with values a_1, a_2 and B_i with values b_1, b_2, with the projection CPTs

$$P\big(A_i = a \mid A_iB_i = a'b'\big) = \big[\, a = a' \,\big], \qquad P\big(B_i = b \mid A_iB_i = a'b'\big) = \big[\, b = b' \,\big].$$

Everything else – the root Ω, the six parameter-set roots, and the lumped node A_iB_i with parents Ω and all six parameter sets – is unchanged.
Asserting the data. Instantiate exactly those of A_i, B_i recorded in d_i: an A-only case becomes $A_i = a_1$, a complete case $A_i = a_1 \wedge B_i = b_2$, an empty case nothing, so D is an ordinary instantiation of a subset of the variables.
Correctness. The new nodes are leaves, so pruning them (Theorem 6.4, Section 6.9.1) leaves the marginal over $\Omega, \theta, A_1B_1, \ldots, A_NB_N$ untouched, and the support of $A_i = a_1$ is exactly the parent instantiations a_1b_1, a_1b_2, so

$$P\big(\cdot \mid A_i = a_1\big) = P\big(\cdot \mid A_iB_i = a_1b_1 \vee A_iB_i = a_1b_2\big).$$

The queries $P(\Omega \mid D)$, $P(\theta \mid D)$ and predictions on A_{N+1}, B_{N+1} are therefore answered for incomplete D as for complete.

17.4 Exercises 18.22–18.24

Problem 18.22. Consider a BDe score that is specified by a prior distribution Pr (defined over the base network variables) together with an equivalent sample size $\psi > 0$, so that the Dirichlet exponent of every parameter $\theta_{x|\mathbf{u}}$ of every candidate structure is

$$\psi_{x|\mathbf{u}} \;=\; \psi \cdot Pr(x, \mathbf{u}),$$

which is Equation 18.25. Let G be a complete network structure, that is, a DAG to which no further edge can be added without creating a directed cycle, and let $\mathbf{d}$ be a complete case, that is, an instantiation of all network variables. Show that $Pr(\mathbf{d}) = P(\mathbf{d}|G)$, where P is the corresponding meta-distribution (the distribution induced by the meta-network of Definition 18.2 built from

structure G together with the Dirichlet priors above).
Note: This exercise provides additional semantics for the prior distribution of a BDe score. Assume Pr is strictly positive, so that every exponent $\psi_{x|\mathbf{u}}$ is positive and every Dirichlet prior is well defined.

Solution. The prior expectations are the prior's conditionals, and completeness turns their product into the chain rule. Write $X_1, \ldots, X_n$ for the variables, $\mathbf{U}_i$ for the parents of X_i in G, and $x_i, \mathbf{u}_i$ for the values $\mathbf{d}$ fixes.
(i) *Prior expectations.* Summing Equation 18.25 gives $\psi_{X|\mathbf{u}} = \psi \cdot Pr(\mathbf{u})$, nonzero since Pr is strictly positive, so by the Dirichlet mean (18.10),

$$\mathrm{Ex}(\theta_{x|\mathbf{u}}) \;=\; \frac{\psi_{x|\mathbf{u}}}{\psi_{X|\mathbf{u}}} \;=\; \frac{\psi \cdot Pr(x, \mathbf{u})}{\psi \cdot Pr(\mathbf{u})} \;=\; Pr(x \mid \mathbf{u}).$$

(ii) *One case under the meta-distribution.* Conditioned on a full parametrization the instance is distributed as the base network, so $P(\mathbf{d} \mid \theta, G) = \prod_i \theta_{x_i|\mathbf{u}_i}$ by Equation 18.20 with $N = 1$. Marginalize against the prior factorization (18.18), valid because the parameter sets are roots and hence independent: the n sets $\theta_{X_i|\mathbf{u}_i}$ are pairwise distinct (distinct variables), so none contributes two factors, while every other set appears only in its own density, which integrates to 1. The remaining n integrals separate into first moments, so with (i),

$$P(\mathbf{d} \mid G) \;=\; \prod_{i=1}^{n} \int \theta_{x_i|\mathbf{u}_i}\, \rho(\theta_{X_i|\mathbf{u}_i})\, d\theta_{X_i|\mathbf{u}_i} \;=\; \prod_{i=1}^{n} Pr(x_i \mid \mathbf{u}_i),$$

for any G.
(iii) *Completeness gives the chain rule.* Relabel so that $X_1, \ldots, X_n$ is a topological order of G. For $i < j$ there must already be an edge between X_i and X_j, since $X_i \to X_j$ respects the order and so could otherwise be added without a cycle; and any such edge runs $X_i \to X_j$, again by the order. Hence $\mathbf{U}_j = \{X_1, \ldots, X_{j-1}\}$ and the instantiation $\mathbf{u}_j$ selected by the complete case $\mathbf{d}$ is $x_1 \ldots x_{j-1}$, so

$$P(\mathbf{d} \mid G) = \prod_{j=1}^{n} Pr(x_j \mid x_1 \ldots x_{j-1}) = Pr(x_1, \ldots, x_n) \;=\; Pr(\mathbf{d}),$$

by the chain rule for Pr and because $\mathbf{d} = x_1 \ldots x_n$.

Problem 18.23. Consider the BDe score over two binary variables A and B, an equivalent sample size of $\psi = 12$, and the prior distribution

A	B	Pr(.)
a_1	b_1	1/4
a_1	b_2	1/6
a_2	b_1	1/4
a_2	b_2	1/3

Recall that the BDe score fixes the Dirichlet exponent of every parameter $\theta_{x|\mathbf{u}}$ by Equation 18.25, $\psi_{x|\mathbf{u}} = \psi \cdot Pr(x, \mathbf{u})$.
Consider also the data set $d_1 = a_1 b_1$, $d_2 = a_1 b_2$ and the structures $G_1 : A \to B$ (a single edge from A to B) and $G_2 : A, B$ (the empty structure, with A and B both roots and no edge between them). Compute the likelihood of each structure using Equation 18.21,

$$P(D) \;=\; \prod_{X\mathbf{U}} \prod_{\mathbf{u}} \frac{\Gamma(\psi_{X|\mathbf{u}})}{\Gamma(\psi_{X|\mathbf{u}} + D\#(\mathbf{u}))} \prod_{x} \frac{\Gamma(\psi_{x|\mathbf{u}} + D\#(x\mathbf{u}))}{\Gamma(\psi_{x|\mathbf{u}})},$$

where $D\#(\alpha)$ is the number of cases in D that satisfy α and $\top$ is the trivial instantiation, so $D\#(\top)$ is the size of the data set.

Solution. $P(D \mid G_1) = 1/26$ and $P(D \mid G_2) = 15/338$, so the empty structure is favoured. With $D\#(\top) = 2$ the counts are

event	count
a_1	2
a_2	0
b_1	1
b_2	1
$a_1 b_1$	1
$a_1 b_2$	1
$a_2 b_1$	0
$a_2 b_2$	0

The prior marginals are

$$Pr(a_1) = \tfrac{1}{4} + \tfrac{1}{6} = \tfrac{5}{12}, \quad Pr(a_2) = \tfrac{1}{4} + \tfrac{1}{3} = \tfrac{7}{12},$$
$$Pr(b_1) = \tfrac{1}{4} + \tfrac{1}{4} = \tfrac{1}{2}, \quad Pr(b_2) = \tfrac{1}{6} + \tfrac{1}{3} = \tfrac{1}{2}.$$

Likelihood of $G_1 : A \to B$. The families are A (empty parent set) and BA, and Equation 18.25 with $\psi = 12$ gives the exponents

parameter	exponent
θ_{a_1}	$12 \cdot \frac{5}{12} = 5$
θ_{a_2}	$12 \cdot \frac{7}{12} = 7$
$\theta_{b_1 \mid a_1}$	$12 \cdot \frac{1}{4} = 3$
$\theta_{b_2 \mid a_1}$	$12 \cdot \frac{1}{6} = 2$
$\theta_{b_1 \mid a_2}$	$12 \cdot \frac{1}{4} = 3$
$\theta_{b_2 \mid a_2}$	$12 \cdot \frac{1}{3} = 4$

with equivalent sample sizes $\psi_A = 12$, $\psi_{B|a_1} = 5$ and $\psi_{B|a_2} = 7$. Equation 18.21 reads

$$\begin{aligned} P(D \mid G_1) = {} & \frac{\Gamma(12)}{\Gamma(12+2)} \frac{\Gamma(5+2)}{\Gamma(5)} \frac{\Gamma(7+0)}{\Gamma(7)} \\ & \cdot \frac{\Gamma(5)}{\Gamma(5+2)} \frac{\Gamma(3+1)}{\Gamma(3)} \frac{\Gamma(2+1)}{\Gamma(2)} \\ & \cdot \frac{\Gamma(7)}{\Gamma(7+0)} \frac{\Gamma(3+0)}{\Gamma(3)} \frac{\Gamma(4+0)}{\Gamma(4)}. \end{aligned}$$

Using $\Gamma(k+m)/\Gamma(k) = k(k+1)\cdots(k+m-1)$ for integers, the three lines evaluate to

$$\begin{aligned} \text{line } 1 &= \frac{1}{12 \cdot 13} \cdot (5 \cdot 6) \cdot 1 = \frac{30}{156} = \frac{5}{26}, \\ \text{line } 2 &= \frac{1}{5 \cdot 6} \cdot 3 \cdot 2 = \frac{6}{30} = \frac{1}{5}, \\ \text{line } 3 &= 1 \cdot 1 \cdot 1 = 1, \end{aligned}$$

so that

$$P(D \mid G_1) \;=\; \frac{5}{26} \cdot \frac{1}{5} \;=\; \frac{1}{26} \;\approx\; 0.03846,$$

the worked example of Section 18.5.2.

Likelihood of $G_2 : A,\ B$. Both families are singletons with empty parent set:

parameter	exponent
θ_{a_1}	$12 \cdot \frac{5}{12} = 5$
θ_{a_2}	$12 \cdot \frac{7}{12} = 7$
θ_{b_1}	$12 \cdot \frac{1}{2} = 6$
θ_{b_2}	$12 \cdot \frac{1}{2} = 6$

with $\psi_A = \psi_B = 12$, the A factor shared with G_1 by parameter modularity. Equation 18.21 gives

$$\begin{aligned} P(D \mid G_2) = {} & \frac{\Gamma(12)}{\Gamma(12+2)} \frac{\Gamma(5+2)}{\Gamma(5)} \frac{\Gamma(7+0)}{\Gamma(7)} \\ & \cdot \frac{\Gamma(12)}{\Gamma(12+2)} \frac{\Gamma(6+1)}{\Gamma(6)} \frac{\Gamma(6+1)}{\Gamma(6)}. \end{aligned}$$

The first line is again $30/156 = 5/26$ and the second is $\frac{1}{12\cdot 13}\cdot 6\cdot 6 = \frac{36}{156} = \frac{3}{13}$, so

$$P(D \mid G_2) \;=\; \frac{5}{26}\cdot\frac{3}{13} \;=\; \frac{15}{338} \;\approx\; 0.04438.$$

Problem 18.24. Consider the BDe score over two binary variables A and B, an equivalent sample size of $\psi = 12$, and the prior distribution

A	B	Pr(.)
a_1	b_1	1/4
a_1	b_2	1/6
a_2	b_1	1/4
a_2	b_2	1/3

so that the Dirichlet exponents are given by Equation 18.25, $\psi_{x|\mathbf{u}} = \psi \cdot Pr(x, \mathbf{u})$.
Consider also the data set $d_1 = a_1b_1$, $d_2 = a_1b_2$ and the structure $G : A \to B$ (a single edge from A to B). Compute the probabilities $P(d_1|G)$, $P(d_2|d_1, G)$ and $P(d_1, d_2|G)$. Compute the same quantities for the structure $G' : B \to A$ (a single edge from B to A) as well.

Solution. All six numbers are $P(d_1) = 1/4$, $P(d_2 \mid d_1) = 2/13$, $P(d_1, d_2) = 1/26$, the same under both structures. Each $P(\cdot \mid G)$ is a marginal likelihood from Equation 18.21, and $P(d_2 \mid d_1, G) = P(d_1, d_2 \mid G)/P(d_1 \mid G)$ by the chain rule. The prior marginals are

$$\begin{aligned} Pr(a_1) &= \tfrac{1}{4}+\tfrac{1}{6} = \tfrac{5}{12}, \quad Pr(a_2) = \tfrac{1}{4}+\tfrac{1}{3} = \tfrac{7}{12},\\ Pr(b_1) &= \tfrac{1}{4}+\tfrac{1}{4} = \tfrac{1}{2}, \quad\;\; Pr(b_2) = \tfrac{1}{6}+\tfrac{1}{3} = \tfrac{1}{2}. \end{aligned}$$

Part 1: $G : A \to B$. The exponents, from $\psi_{x|\mathbf{u}} = 12\,Pr(x, \mathbf{u})$, are $\psi_{a_1} = 5$, $\psi_{a_2} = 7$ (so $\psi_A = 12$); $\psi_{b_1|a_1} = 3$, $\psi_{b_2|a_1} = 2$ (so $\psi_{B|a_1} = 5$); and $\psi_{b_1|a_2} = 3$, $\psi_{b_2|a_2} = 4$ (so $\psi_{B|a_2} = 7$).
For $D = \{d_1\} = \{a_1b_1\}$ the counts are $D\#(\top) = 1$, $D\#(a_1) = 1$, $D\#(a_2) = 0$, $D\#(a_1b_1) = 1$, and all remaining family counts are 0. Equation 18.21 gives

$$\begin{aligned} P(d_1 \mid G) &= \frac{\Gamma(12)}{\Gamma(13)}\frac{\Gamma(5+1)}{\Gamma(5)}\frac{\Gamma(7)}{\Gamma(7)}\\ &\quad\cdot\frac{\Gamma(5)}{\Gamma(6)}\frac{\Gamma(3+1)}{\Gamma(3)}\frac{\Gamma(2)}{\Gamma(2)}\cdot\frac{\Gamma(7)}{\Gamma(7)}\cdot 1\cdot 1\\ &= \frac{5}{12}\cdot\frac{3}{5} \;=\; \frac{1}{4}, \end{aligned}$$

which is $Pr(a_1b_1)$, as Exercise 18.22 requires of a complete DAG on a complete case. For $D = \{d_1, d_2\}$ the counts become $D\#(\top) = 2$, $D\#(a_1) = 2$, $D\#(a_1b_1) = D\#(a_1b_2) = 1$ with the rest 0, and Equation 18.21 gives $P(d_1, d_2 \mid G) = \frac{30}{156}\cdot\frac{6}{30} = \frac{1}{26}$, the computation of Exercise 18.23. Therefore

$$P(d_2 \mid d_1, G) \;=\; \frac{1/26}{1/4} \;=\; \frac{4}{26} \;=\; \frac{2}{13} \;\approx\; 0.1538.$$

Part 2: $G' : B \to A$. The families are B (root) and AB, so Equation 18.25 with the same Pr and $\psi = 12$ gives exponents $\psi_{b_1} = 12\cdot\frac{1}{2} = 6$, $\psi_{b_2} = 6$ (so $\psi_B = 12$); $\psi_{a_1|b_1} = 12\cdot\frac{1}{4} = 3$, $\psi_{a_2|b_1} = 12\cdot\frac{1}{4} = 3$ (so $\psi_{A|b_1} = 6$); and $\psi_{a_1|b_2} = 12\cdot\frac{1}{6} = 2$, $\psi_{a_2|b_2} = 12\cdot\frac{1}{3} = 4$ (so $\psi_{A|b_2} = 6$).
For $D = \{d_1\} = \{a_1b_1\}$: $D\#(\top) = 1$, $D\#(b_1) = 1$, $D\#(b_2) = 0$, $D\#(a_1b_1) = 1$, all others 0. Then

$$\begin{aligned} P(d_1 \mid G') &= \frac{\Gamma(12)}{\Gamma(13)}\frac{\Gamma(6+1)}{\Gamma(6)}\frac{\Gamma(6)}{\Gamma(6)}\\ &\quad\cdot\frac{\Gamma(6)}{\Gamma(7)}\frac{\Gamma(3+1)}{\Gamma(3)}\frac{\Gamma(3)}{\Gamma(3)}\cdot 1\\ &= \frac{6}{12}\cdot\frac{3}{6} \;=\; \frac{1}{4}, \end{aligned}$$

again $Pr(a_1 b_1)$, since $B \to A$ is also a complete DAG. For $D = \{d_1, d_2\}$: $D\#(\top) = 2$, $D\#(b_1) = 1$, $D\#(b_2) = 1$, $D\#(a_1 b_1) = 1$, $D\#(a_2 b_1) = 0$, $D\#(a_1 b_2) = 1$, $D\#(a_2 b_2) = 0$. Equation 18.21 gives

$$\begin{aligned} P(d_1, d_2 \mid G') = & \frac{\Gamma(12)}{\Gamma(14)} \frac{\Gamma(6+1)}{\Gamma(6)} \frac{\Gamma(6+1)}{\Gamma(6)} \\ & \cdot \frac{\Gamma(6)}{\Gamma(7)} \frac{\Gamma(3+1)}{\Gamma(3)} \frac{\Gamma(3+0)}{\Gamma(3)} \\ & \cdot \frac{\Gamma(6)}{\Gamma(7)} \frac{\Gamma(2+1)}{\Gamma(2)} \frac{\Gamma(4+0)}{\Gamma(4)}. \end{aligned}$$

The three lines evaluate to

$$\begin{aligned} \text{line } 1 &= \frac{1}{12 \cdot 13} \cdot 6 \cdot 6 = \frac{36}{156} = \frac{3}{13}, \\ \text{line } 2 &= \frac{1}{6} \cdot 3 \cdot 1 = \frac{1}{2}, \\ \text{line } 3 &= \frac{1}{6} \cdot 2 \cdot 1 = \frac{1}{3}, \end{aligned}$$

so that

$$P(d_1, d_2 \mid G') \;=\; \frac{3}{13} \cdot \frac{1}{2} \cdot \frac{1}{3} \;=\; \frac{1}{26},$$

and consequently

$$P(d_2 \mid d_1, G') \;=\; \frac{1/26}{1/4} \;=\; \frac{2}{13}.$$

Summary.

quantity	$G : A \to B$	$G' : B \to A$
$P(d_1)$	$1/4$	$1/4$
$P(d_2 \mid d_1)$	$2/13$	$2/13$
$P(d_1, d_2)$	$1/26$	$1/26$

The agreement is the likelihood equivalence that names the BDe score: $A \to B$ and $B \to A$ are Markov equivalent complete DAGs over $\{A, B\}$, so the BDe assumptions force equal marginal likelihoods on every complete data set, and the conditionals agree as ratios of those.