

Bayesian Networks

Aayush Bajaj

August 31, 2026

a joint probability distribution over n binary variables is a table with $2^n - 1$ free numbers. at $n = 30$ that is a billion parameters for a toy problem — nobody can elicit them, store them, or estimate them from data. the bayesian network is the classical answer: encode the *independencies* of the distribution in a directed acyclic graph, and the joint collapses into a product of small local tables. the graph is simultaneously a data structure for storage, a calculus for reading off independence, and the substrate on which every inference algorithm runs.

the treatment here follows darwiche's book and unsw's comp9418, which is taught from it

Contents

1	from joint distributions to graphs	1
2	definition	2
2.1	the sprinkler network	2
2.2	how many parameters?	3
3	independence semantics	3
3.1	d-separation and the three valves	3
3.2	testing d-separation in linear time	4
3.3	soundness, completeness, i-maps	4
3.4	markov blankets	4
4	bayesian networks as classifiers	5
4.1	naive bayes	5
4.2	tree-augmented naive bayes	5
5	gaussian bayesian networks	5
6	a note on causality	6
7	results	6
8	see also	6

1 from joint distributions to graphs

everything in probabilistic reasoning reduces to operations on the joint distribution $P(X_1, \dots, X_n)$: marginalise out what you don't care about, condition on what you observed. the problem is purely representational — the joint is exponential in n .

the chain rule factorises any joint, with no assumptions, in any variable order:

$$P(x_1, \dots, x_n) = \prod_{i=1}^n P(x_i \mid x_1, \dots, x_{i-1}). \quad (1)$$

this trades one big table for n conditional tables, the last of which is just as big as the joint — no saving yet. the saving comes from *conditional independence*: if X_i is independent of most of its predecessors given a few of them (its parents $\mathbf{U}_i$), each factor shrinks to $P(x_i \mid \mathbf{u}_i)$. sparse independence structure is what a bayesian network stores; the graph is nothing but a bookkeeping device for which conditioning variables survived.

*sparser graph $\rightarrow$ fewer
parameters $\rightarrow$ cheaper
inference: eliminating edges is
the whole game*

2 definition

a **bayesian network** over variables $X_1, \dots, X_n$ is a pair (G, Θ) :

- G — a directed acyclic graph with one node per variable;
- Θ — for each variable X with parents $\mathbf{U}$, a **conditional probability table** (cpt) $\theta_{x|\mathbf{u}} = P(x \mid \mathbf{u})$, one distribution over X per parent instantiation $\mathbf{u}$.

the semantics of the graph are the **markovian assumptions**: every variable is independent of its non-descendants given its parents,

$$X \perp \text{NonDescendants}(X) \mid \text{Parents}(X), \quad (2)$$

written collectively as $\text{Markov}(G)$. from these assumptions and the ordinary chain rule follows the **chain rule for bayesian networks**:

$$P(x_1, \dots, x_n) = \prod_{i=1}^n \theta_{x_i|\mathbf{u}_i}, \quad (3)$$

the joint is the product of one cpt entry per variable. this factorisation is the network's entire content: a bayesian network *is* a compact representation of a unique joint distribution.

2.1 the sprinkler network

the running example of the course (and of darwiche's book): winter makes rain likely and sprinkler use unlikely; either wets the grass; rain alone makes the road slippery.

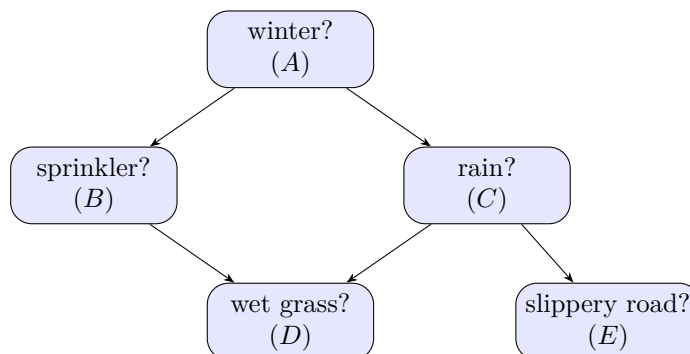


Figure 1: the sprinkler network. each node carries a cpt conditioned on its parents; the five small tables replace a joint over $2^5 = 32$ states.

the parametrisation, writing a for $A = \text{true}$ and $\bar{a}$ for $A = \text{false}$:

cpt	entries
θ_A	$P(a) = 0.6$
$\theta_{B A}$	$P(b a) = 0.2, P(b \bar{a}) = 0.75$
$\theta_{C A}$	$P(c a) = 0.8, P(c \bar{a}) = 0.1$
$\theta_{D B,C}$	$P(d b, c) = 0.95, P(d b, \bar{c}) = 0.9, P(d \bar{b}, c) = 0.8, P(d \bar{b}, \bar{c}) = 0$
$\theta_{E C}$	$P(e c) = 0.7, P(e \bar{c}) = 0$

only the true-rows need storing — each row of a cpt is a distribution and must sum to one, so the false-entries are complements. the chain rule then prices any world instantly:

a useful sanity check when eliciting tables: fix every parent instantiation and confirm the child's entries sum to 1

$$P(a, \bar{b}, c, d, e) = \theta_a \theta_{\bar{b}|a} \theta_{c|a} \theta_{d|\bar{b},c} \theta_{e|c} = (0.6)(0.8)(0.8)(0.8)(0.7) = 0.21504. \quad (4)$$

2.2 how many parameters?

for a variable with d states and k parents of d states each, the cpt has $O(d^{k+1})$ entries — exponential in the *in-degree*, not in n . a network of n variables with maximum in-degree k costs $O(n d^{k+1})$ parameters total. the moral: what makes a bayesian network compact is bounded fan-in, and when a node unavoidably has many parents, structured cpt representations (noisy-or, decision trees, rules) replace the exponential table.

3 independence semantics

the markovian assumptions are only the *axioms*. many more independencies follow from them, and the **graphoid axioms** generate them: symmetry, decomposition, weak union, contraction, and (for strictly positive distributions) intersection. deriving independencies by hand from these axioms is tedious; the point of the graph is that a purely graphical test — d-separation — captures their full inferential power.

3.1 d-separation and the three valves

think of every path between two nodes as a pipe and every intermediate node as a **valve**. a valve is either open (lets dependence flow) or closed (blocks it), and its behaviour depends on how the path's edges meet at it:

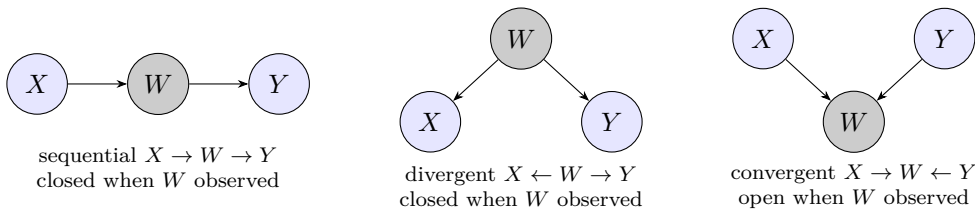


Figure 2: the three valve types. shading marks a node in the conditioning set $\mathbf{Z}$: observing a sequential or divergent node closes it; observing a convergent node (or any of its descendants) opens it.

- **sequential** (chain) $X \rightarrow W \rightarrow Y$: open by default, closed once W is observed — knowing the intermediate cause screens off the ends. in the sprinkler network, $A \rightarrow C \rightarrow E$: once you know it rained, winter tells you nothing more about the road.
- **divergent** (common cause) $X \leftarrow W \rightarrow Y$: open by default, closed once W is observed. $B \leftarrow A \rightarrow C$: sprinkler use and rain correlate only through the season.

- **convergent** (common effect, the collider) $X \rightarrow W \leftarrow Y$: *closed* by default, *opened* by observing W or any descendant of W . $B \rightarrow D \leftarrow C$: sprinkler and rain are independent a priori, but once the grass is seen wet they compete — learning it rained “explains away” the sprinkler.

explaining away is the one behaviour with no analogue in undirected models, and the reason colliders get their own rule

a path is **blocked** by $\mathbf{Z}$ if at least one of its valves is closed. disjoint sets $\mathbf{X}$ and $\mathbf{Y}$ are **d-separated** by $\mathbf{Z}$, written $\text{dsep}_G(\mathbf{X}, \mathbf{Z}, \mathbf{Y})$, iff every path from $\mathbf{X}$ to $\mathbf{Y}$ is blocked by $\mathbf{Z}$.

3.2 testing d-separation in linear time

enumerating paths is exponential; the practical test is a graph surgery followed by plain reachability:

this is precisely the algorithm implemented for the comp9418 assignment — prune, delete, then one bfs/dfs

1. **prune** — repeatedly delete any leaf node not in $\mathbf{X} \cup \mathbf{Y} \cup \mathbf{Z}$. (an unobserved leaf is a closed convergent valve or an inferentially inert endpoint; removing it can close nothing that was open.)
2. **delete** — remove every edge leaving a node in $\mathbf{Z}$. (with sequential and divergent valves closed by observation, only inbound edges of $\mathbf{Z}$ nodes — the open collider case — survive.)
3. **test** — $\mathbf{X}$ and $\mathbf{Y}$ are d-separated iff they are disconnected in the surgered graph, checkable by depth-first search.

the whole procedure is $O(n + m)$ in the size of the graph.

3.3 soundness, completeness, i-maps

- **soundness**: if $\text{dsep}_G(\mathbf{X}, \mathbf{Z}, \mathbf{Y})$ then $\mathbf{X} \perp \mathbf{Y} \mid \mathbf{Z}$ in *every* distribution that factorises over G . d-separation never lies about independence.
- **incompleteness**: the converse fails — a particular parametrisation may hide extra independencies the graph cannot see (numbers can conspire; e.g. a cpt that ignores one of its parents). so connectedness does *not* prove dependence.

a dag G is an **i-map** (independence map) of a distribution P if every independence claimed by G holds in P — the factorisation guarantees exactly this. it is a **minimal i-map** if no edge can be removed without breaking the property, and a **p-map** (perfect map) if additionally every independence in P is visible as a d-separation. not every distribution has a p-map; the fully connected dag is a trivial (useless) i-map of everything, which is why “sparser is better” has formal content.

3.4 markov blankets

the **markov blanket** of X is its parents, its children, and its children’s other parents (spouses). conditioned on its blanket, X is independent of everything else in the network:

$$X \perp \mathbf{V} \setminus (\{X\} \cup \text{MB}(X)) \mid \text{MB}(X). \quad (5)$$

the spouses are the non-obvious members: observing a common child opens the collider, so the other parent becomes relevant and must be conditioned on to seal X off. the blanket is the theoretical basis of gibbs sampling (each resampling step conditions only on the blanket) and a handy feature-selection notion: the blanket of a class variable is the minimal set of observations worth making.

4 bayesian networks as classifiers

4.1 naive bayes

put the class C on top, one arrow to each attribute:

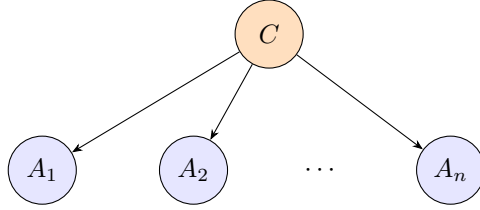


Figure 3: naive bayes: attributes are conditionally independent given the class. parameters grow linearly in the number of attributes.

$$P(C, A_1, \dots, A_n) = P(C) \prod_{i=1}^n P(A_i | C), \quad (6)$$

which needs $O(|C| + n|C||A|)$ parameters instead of $O(|C||A|^n)$ — linear in n . classification is a maximum a posteriori query, $\hat{c} = \arg \max_c P(c) \prod_i P(a_i | c)$, computed in log-space to dodge underflow when n is large (a bag-of-words spam filter multiplies thousands of tiny probabilities). the assumption that attributes are independent given the class is usually false, which makes naive bayes systematically *over-confident*: its posteriors saturate towards 0 and 1. fine for picking the argmax, dangerous when the probability itself matters (medical diagnosis).

4.2 tree-augmented naive bayes

tan relaxes the independence assumption one notch: on top of the class arrows, the attributes are allowed a tree of edges among themselves, chosen to maximise conditional mutual information $I(A_i; A_j | C)$ — a chow-liu construction. each attribute then has at most two parents (the class and one attribute), so parameters stay small while the worst naive-bayes violations get patched. further relaxations (ban, gbn) trade more structure for more data hunger.

5 gaussian bayesian networks

continuous variables break the table representation, but the machinery survives if every cpt is a **linear gaussian** model:

$$X | \mathbf{u} \sim \mathcal{N}(\beta_0 + \boldsymbol{\beta}^\top \mathbf{u}, \sigma^2), \quad (7)$$

the child's mean is linear in its parents and its variance is constant. a network whose cpds are all linear gaussian defines a joint multivariate gaussian, and vice versa — every multivariate gaussian factorises as a gaussian bayesian network in any variable ordering. inference stays closed-form (conditioning and marginalising gaussians yields gaussians), which is what makes kalman filters exact, and it is the setting for gaussian belief propagation on the approximate inference page.

the correspondence is constructive: means and covariances fall out of the betas by forward substitution, and independencies hide in the zeros of the precision matrix

6 a note on causality

the arrows of a bayesian network are *not* causal claims — any dag consistent with the distribution’s independencies is an equally valid i-map, and reversing edges (with recomputed cpts) can represent the same joint. that said, networks built along causal directions are the ones humans can elicit (“rain causes wet grass” has a natural cpt; the reverse does not), tend to be sparsest, and are the only ones that answer *interventional* questions correctly: setting $\text{do}(D = d)$ — forcing the grass wet — severs D from its parents and, unlike conditioning on $D = d$, licenses no inference back up to rain. observation flows both ways along edges; intervention only flows down the causal ones.

7 results

curated statements worth having at your fingertips:

- **chain rule for bayesian networks** — if P satisfies $\text{Markov}(G)$, then $P(x_1, \dots, x_n) = \prod_i \theta_{x_i | \mathbf{u}_i}$.
- **parameter complexity** — a network of n variables, each with $\leq d$ states and $\leq k$ parents, has $O(n d^{k+1})$ parameters.
- **soundness of d-separation** — $\text{dsep}_G(\mathbf{X}, \mathbf{Z}, \mathbf{Y}) \implies \mathbf{X} \perp \mathbf{Y} \mid \mathbf{Z}$ for every P factorising over G .
- **incompleteness of d-separation** — there exist G, P with P faithful to the factorisation where $\mathbf{X} \perp \mathbf{Y} \mid \mathbf{Z}$ holds yet $\mathbf{X}, \mathbf{Y}$ are d-connected.
- **d-separation complexity** — decidable in $O(n+m)$ by pruning + edge deletion + reachability.
- **markov blanket** — $\text{MB}(X) = \text{Parents}(X) \cup \text{Children}(X) \cup \text{Spouses}(X)$ renders X independent of the rest of the network.
- **gaussian correspondence** — linear gaussian networks $\equiv$ multivariate gaussians.

8 see also

- exact inference — variable elimination and the jointree algorithm on these networks
- approximate inference — sampling when exact is too expensive
- markov networks — the undirected cousins
- markov chains and hidden markov models — bayesian networks unrolled over time
- learning graphical models — estimating the cpts and the structure from data
- modeling and reasoning with bayesian networks — darwiche’s book, the canonical reference for this material