

Curse of Dimensionality

Aayush Bajaj

July 10, 2026

Contents

1	motivation	1
2	volume hides at the boundary	2
2.1	the epsilon-shell of the cube	2
2.2	the ball loses to the cube	2
3	neighbourhoods stop being local	2
4	distances concentrate	3
4.1	nearest neighbours drift away	3
4.2	all distances become the same distance	3
4.3	sampling cannot keep up	3
5	a monte-carlo audit	4
6	what it does to knn and kernel methods	4
7	why learning still works	5
8	see also	5

geometric intuition is trained in $p \leq 3$ and it does not survive the trip upstairs. in high dimensions the volume of a cube hides in its corners, every point is near the boundary, all pairwise distances look alike, and “local” neighbourhoods must stretch almost the full width of the space before they contain any data. every method that reasons from *closeness* — knn, kernel smoothers, rbf kernels — inherits these pathologies at once.

the phrase is bellman's, coined in the preface of his 1957 /dynamic programming/, where filling a grid over a high-dimensional state space first went bankrupt

1 motivation

- local methods (see [the bias-variance decomposition](#)) work by a simple bargain: approximate $f(x_0)$ by averaging the targets of training points *near* x_0 . the bargain assumes near points exist and that “near” is informative.
- both assumptions are functions of the dimension p , and both fail — quantitatively, not vaguely — as p grows. this page collects the four standard calculations, following ch. 2.5 of (Hastie, Trevor and Tibshirani, Robert and Friedman, Jerome, 2009), then audits them by monte carlo.

2 volume hides at the boundary

2.1 the epsilon-shell of the cube

take the unit hypercube $[0, 1]^p$ and peel off a shell of thickness ε from every face. the interior that remains is a cube of edge $1 - 2\varepsilon$, so the fraction of volume *within* ε of the boundary is

$$\text{shell}_p(\varepsilon) = 1 - (1 - 2\varepsilon)^p \xrightarrow{p \rightarrow \infty} 1. \quad (1)$$

with $\varepsilon = 0.05$: 19% of the square, 41% of the 5-cube, 65% of the 10-cube, 99.5% of the 50-cube. uniform data in high dimension is *all crust, no crumb* — and prediction near the boundary is extrapolation, not interpolation: your neighbours lie to one side (Hastie, Trevor and Tibshirani, Robert and Friedman, Jerome, 2009).

2.2 the ball loses to the cube

the unit-radius p -ball has volume $V_p = \pi^{p/2}/\Gamma(p/2 + 1)$, which itself tends to zero; worse, the ball *inscribed in* the unit cube (radius $\frac{1}{2}$) occupies $V_p 2^{-p}$ of it — about 0.25% at $p = 10$. almost none of a high-dimensional cube is anywhere near its centre; almost all of it is corner.

meanwhile the centre-to-corner distance $\sqrt{p}/2$ grows without bound while centre-to-face stays 1/2: the cube becomes a spiky ball with 2^p long thin horns

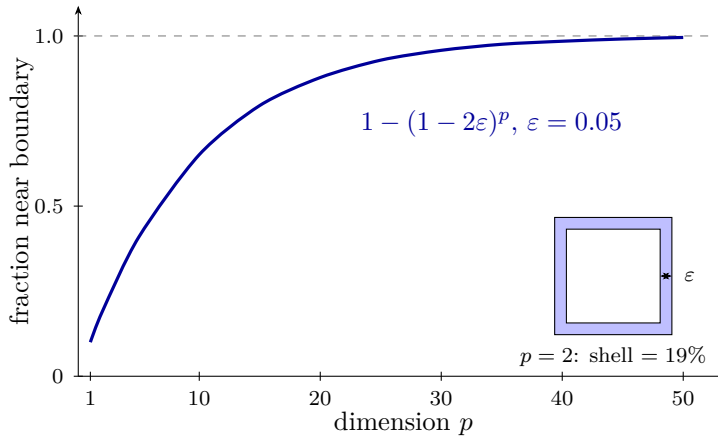


Figure 1: fraction of the unit cube within $\varepsilon = 0.05$ of its boundary, $1 - (1 - 2\varepsilon)^p$. by $p = 50$ the cube is essentially all boundary. inset: the $p = 2$ case — the shaded shell is already 19% of the area.

3 neighbourhoods stop being local

suppose data are uniform on $[0, 1]^p$ and you want a hypercubical neighbourhood around a query point that captures a fraction r of the observations. matching volumes, its edge length must be

$$e_p(r) = r^{1/p}, \quad (2)$$

the calculation behind figure 2.6 of (Hastie, Trevor and Tibshirani, Robert and Friedman, Jerome, 2009). plug in numbers for $p = 10$: capturing just 1% of the data needs $e_{10}(0.01) \approx 0.63$ — the “neighbourhood” spans 63% of the range of *every* coordinate. capturing 10% needs $e_{10}(0.1) \approx 0.80$. such neighbourhoods are not local

in any useful sense, and shrinking them to restore locality leaves them empty — the variance side of the trade explodes instead.

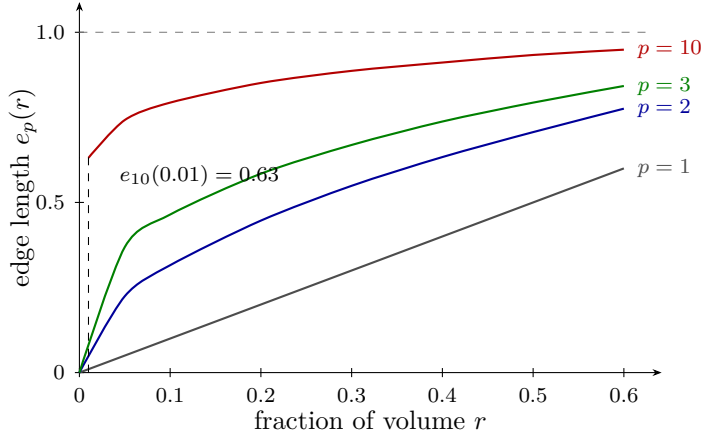


Figure 2: edge length $e_p(r) = r^{1/p}$ of the sub-cube needed to capture a fraction r of uniformly distributed data (after ESL fig. 2.6). at $p = 10$, even a 1% neighbourhood spans 63% of each coordinate.

4 distances concentrate

4.1 nearest neighbours drift away

for n points uniform in the unit ball in $\mathbb{R}^p$, the median distance from the origin to its *nearest* neighbour has the closed form

$$d(p, n) = \left(1 - \frac{1}{2}^{1/n}\right)^{1/p}, \quad (3)$$

eq. (2.24) of (Hastie, Trevor and Tibshirani, Robert and Friedman, Jerome, 2009). for $n = 500$, $p = 10$: $d \approx 0.52$ — the nearest neighbour is typically *more than halfway to the boundary*. the “nearest” point is not near, and its target value carries little information about f at the query.

4.2 all distances become the same distance

concentration cuts deeper: not only do nearest neighbours drift outward, the *contrast* between near and far evaporates. two complementary facts:

- for $x \sim \mathcal{N}(0, I_p)$, $\mathbb{E}\|x\|^2 = p$ and $\text{Var}(\|x\|)$ stays $O(1)$, so the norm concentrates in a thin shell around $\sqrt{p}$: relative spread $\text{sd}(\|x\|)/\mathbb{E}\|x\| = O(1/\sqrt{p})$. high-dimensional gaussians are soap bubbles, not fuzzy balls.
- for i.i.d. points with i.i.d. coordinates, pairwise squared distances are sums of p i.i.d. terms, so by the law of large numbers $\|x - x'\|^2/p$ converges to a constant: the ratio $\min_{ij} d_{ij}/\max_{ij} d_{ij} \rightarrow 1$ as $p \rightarrow \infty$ with n fixed. beyer et al. (1999, *when is “nearest neighbor” meaningful?*) formalise this as relative contrast $(d_{\max} - d_{\min})/d_{\min} \rightarrow 0$: the nearest-neighbour *ranking* itself becomes noise-dominated.

this is why ‘distance to the mean’ is a useless anomaly score in high dimension — every sample is roughly equally far from it

4.3 sampling cannot keep up

the only cure uniform geometry accepts is data, and the bill is exponential. if n_1 samples give adequate density in one dimension, matching that density in p

dimensions costs n_1^p samples — equivalently, the density achieved by n points in p dimensions scales as $n^{1/p}$ (Hastie, Trevor and Tibshirani, Robert and Friedman, Jerome, 2009). with $n_1 = 100$ and $p = 10$, you owe 10^{20} samples. no dataset pays that; every practical high-dimensional method is betting on structure instead (see below).

5 a monte-carlo audit

all four phenomena in one table: boundary crowding, nearest-neighbour drift, min/max contrast collapse, and gaussian shell concentration.

```

1 import numpy as np
2
3 rng = np.random.default_rng(0)
4 n = 1000                                # points per experiment
5
6 print(f"{'p':>4} {'frac near bdry':>14} {'med nn dist':>11} "
7       f"{'min/max dist':>12} {'sd/mean |g|':>11}")
8 for p in (1, 2, 5, 10, 20, 50, 100):
9     x = rng.uniform(0, 1, (n, p))        # uniform on the unit hypercube
10    # fraction of points within 0.05 of the boundary
11    near = np.mean((x.min(axis=1) < 0.05) | (x.max(axis=1) > 0.95))
12    # pairwise distances: nearest-neighbour and min/max concentration
13    d = np.linalg.norm(x[:, None, :] - x[None, :, :], axis=-1)
14    np.fill_diagonal(d, np.inf)
15    med_nn = np.median(d.min(axis=1))
16    off = d[np.isfinite(d)]
17    ratio = off.min() / off.max()
18    # relative spread of the norm of a standard gaussian
19    r = np.linalg.norm(rng.normal(0, 1, (n, p)), axis=1)
20    print(f"{'p':>4} {'near:>14.3f} {'med_nn:>11.3f} {'ratio:>12.4f} "
21          f"{'r.std() / r.mean():>11.4f}")

```

p	frac near bdry	med nn dist	min/max dist	sd/mean g
1	0.093	0.000	0.0000	0.7358
2	0.184	0.015	0.0003	0.5076
5	0.410	0.180	0.0140	0.3370
10	0.615	0.500	0.1298	0.2204
20	0.884	1.051	0.2518	0.1546
50	0.994	2.128	0.4475	0.0958
100	1.000	3.337	0.5814	0.0725

reading the columns: boundary fraction tracks $1 - 0.9^p$ (theory: 0.65 at $p = 10$, 0.995 at $p = 50$); with $n = 1000$ fixed, the median nearest-neighbour distance grows past 3 while the *whole cube* has unit edge; the min/max ratio climbs toward 1; the gaussian norm's relative spread decays like $1/\sqrt{2p}$.

6 what it does to knn and kernel methods

- **knn**: the estimator $\hat{f}(x_0) = \frac{1}{k} \sum_{\ell} y_{(\ell)}$ is built on the premise that $f(x_{(\ell)}) \approx f(x_0)$. when the k -th neighbour sits at distance $\Theta(1)$ regardless of k , that premise fails and the *bias* term of the decomposition swallows the estimate — ESL's worked example with $f(x) = e^{-8\|x\|^2}$ shows 1-nn bias alone approaching the full signal amplitude by $p = 10$ (Hastie, Trevor and Tibshirani, Robert and Friedman, Jerome, 2009). and because the nn ranking loses contrast, even the identity of the neighbours becomes arbitrary.

- **kernel smoothers and rbf kernels:** a bandwidth- h kernel is a soft neighbourhood, so the same dichotomy applies — small h means empty neighbourhoods (variance), large h means global averaging (bias). in the kernel-machine setting (see [kernel methods](#)), concentration of distances flattens the gram matrix of an rbf kernel toward a constant matrix plus identity: all off-diagonal similarities converge to the same value, and the kernel stops discriminating.
- **density estimation and histograms:** a histogram with b bins per axis has b^p cells; almost all are empty at any feasible n . density estimation is the purest victim — this is the standard motivation for dimensionality reduction before density modelling (Deisenroth, Marc Peter and Faisal, A. Aldo and Ong, Cheng Soon, 2020).
- **rates:** nonparametric minimax rates for estimating a lipschitz-smooth f degrade as $n^{-1/(p+2)}$ -type exponents — to halve the error you must *exponentiate* the sample size in p (Wasserman, Larry, 2010).

7 why learning still works

the curse is a statement about *uniform* measure and *generic* targets; real data decline to be either.

- **the manifold hypothesis.** images, speech, text embeddings occupy low-dimensional structures inside the ambient $\mathbb{R}^p$; effective dimension, not ambient dimension, sets the rates. a face dataset in 10^6 pixels does not fill $[0, 1]^{10^6}$ — it traces a surface of perhaps a few hundred intrinsic dimensions.
- **structured function classes.** additive models, sparse linear models, and convolutional hierarchies each restrict f so estimation error depends on the structure’s complexity, not on p . this is the same purchase of inductive bias that [the no free lunch theorem](#) says you must make anyway.
- **blessings of dimensionality.** concentration itself is a tool: random projections nearly preserve pairwise distances (johnson–lindenstrauss), gaussians become effectively spherical shells, and high-dimensional random vectors are nearly orthogonal — properties that hashing, sketching, and overparameterised models exploit.

the practical checklist: reduce dimension when the structure is linear-ish (pca and friends), regularise hard, prefer models whose bias matches your domain, and never trust raw euclidean distance in more than a few dozen dimensions without checking the contrast.

8 see also

- [the bias-variance decomposition](#) — the curse is its p -dependence made explicit
- [the no free lunch theorem](#) — why structure assumptions are the only exit
- [kernel methods](#) — rbf similarity and its high-dimensional flattening
- [cross validation](#) — measuring how badly the curse is biting your model

references

Deisenroth, Marc Peter and Faisal, A. Aldo and Ong, Cheng Soon (2020). *Mathematics for Machine Learning*, Cambridge University Press.

Hastie, Trevor and Tibshirani, Robert and Friedman, Jerome (2009). *The Elements of Statistical Learning*, Springer.

Wasserman, Larry (2010). *All of Statistics: A Concise Course in Statistical Inference*, Springer.