Introduction to Data Science — Chapters 4-10

Risk, Estimation, RNG, Markov Chains, Pattern Recognition, High Dimension, Dimensionality Reduction

Benny Avelin

Uppsala University

8th December 2025

Chapter overview 4-10

1. **Risk & supervised learning (Ch. 4)**
2. **Estimation, bias-variance, CIs (Ch. 5)**
3. **Random number generation & sampling (Ch. 6)**
4. **Markov chains & convergence (Ch. 7)**
5. **Pattern recognition & classification (Ch. 8)**
6. **High-dimensional phenomena (Ch. 9)**
7. **Dimensionality reduction & PCA (Ch. 10)**

Ch.4 Risk: supervised learning formalism I

Modeling setup (statistical model)

- **Statistical model:** a family of joint distributions $\mathcal{F} = \{F_\theta : \theta \in \Theta\}$ for $Z = (X, Y)$.
- Generator: $X \sim F_X$. Supervisor: $Y \mid X \sim F_{Y|X}$. Data: $Z = (X, Y) \sim F \in \mathcal{F}$.

Risk and learning machine

- **Model class (hypothesis space)** $\mathcal{M}$: set of predictors $g : \mathcal{X} \to \mathcal{Y}$.
- **Risk:** $R(g) = \mathbb{E}[L(Z, g)]$, want a *learning machine*

$$\mathcal{A} : \ (Z_1, \dots, Z_n) \mapsto \hat{g}_n \in \mathcal{M}$$

  that (approximately) minimizes $R(g)$ over $g \in \mathcal{M}$.
- **Regression target:** $r(x) = \mathbb{E}[Y \mid X = x]$.
- **Bayes classifier (binary):** $h^*(x) = \mathbf{1}\{\eta(x) > \frac{1}{2}\}$ with $\eta(x) = \mathbb{P}(Y = 1 \mid X = x)$; minimizes 0-1 risk.

Ch.4 Risk: supervised learning formalism II

ERM / log-loss connection

- **ERM learning machine:**

$$\hat{g}_n \in \arg\min_{g\in\mathcal{M}} \hat{R}_n(g), \qquad \hat{R}_n(g) = \frac{1}{n}\sum_{i=1}^{n} L(Z_i, g).$$

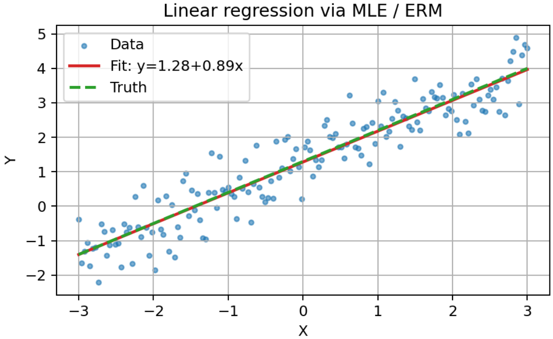
- Parametric family $\{p_\alpha\}$, loss $L(z,\alpha) = -\log p_\alpha(z) \Rightarrow \arg\min_\alpha \hat{R}_n(p_\alpha)$ is the *MLE*.

Ch.4 Linear regression from conditional Gaussian

Assume $Y \mid X = x \sim \mathcal{N}(ax + b, \sigma^2)$ and f_X fixed. Negative log-likelihood (NLL):

$$\mathcal{L}(a, b, \sigma) = \sum_{i=1}^{n} \left(\tfrac{1}{2} \log \sigma^2 + \tfrac{(y_i - (ax_i + b))^2}{2\sigma^2} \right) + \text{const.}$$

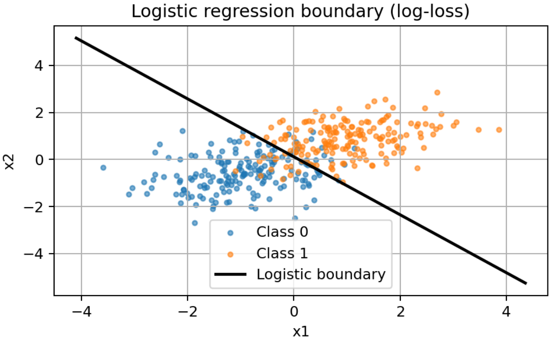
For fixed σ, minimizing w.r.t. (a, b) equals minimizing $\sum_i (y_i - (ax_i + b))^2$ *(ordinary least squares)*.



Ch.4 Logistic regression details

Model $\mathbb{P}(Y = 1 \mid x) = \sigma(w^\top x)$, $\sigma(t) = 1/(1 + e^{-t})$.

- NLL: $\ell(w) = \sum_{i=1}^{n} \left(\log(1 + e^{-y_i w^\top x_i})\right)$ with $y_i \in \{\pm 1\}$.
- Gradient: $\nabla \ell(w) = \sum_i \left(\sigma(-y_i w^\top x_i)(-y_i)x_i\right)$; Hessian: $\sum_i \sigma(\cdot)(1 - \sigma(\cdot))x_i x_i^\top \succeq 0$ (convex).
- Regularization (ridge): add $\frac{\lambda}{2}\|w\|^2$ to control variance and improve generalization.



Ch.5 Estimation: estimators and bias-variance-MSE

- **Estimator:** a rule $\hat{\Theta}_n = g(X_1, \ldots, X_n)$ used to guess an unknown parameter θ^*. Examples: sample mean, sample variance, MLE.
- **Bias and variance:**

$$\text{bias}(\hat{\Theta}_n) = \mathbb{E}[\hat{\Theta}_n] - \theta^*, \qquad \text{se}(\hat{\Theta}_n) = \sqrt{\text{Var}(\hat{\Theta}_n)}.$$

For squared-error loss:

$$\mathbb{E}\left[(\hat{\Theta}_n - \theta^*)^2\right] = \text{bias}^2(\hat{\Theta}_n) + \text{se}^2(\hat{\Theta}_n) \quad \text{(MSE)}.$$

Ch.5 Estimation: consistency and efficiency

- **Consistency (LLN/CLT):** for i.i.d. with mean μ and var. σ^2:

$$\bar{X}_n = \frac{1}{n}\sum_{i=1}^{n} X_i \xrightarrow{p} \mu \quad \text{(LLN: estimator converges to truth)},$$

$$\sqrt{n}(\bar{X}_n - \mu) \Rightarrow \mathcal{N}(0, \sigma^2) \quad \text{(CLT: approximate sampling distribution)}.$$

- **Unbiased vs efficient:** e.g. variance estimators

$$S_n^2 = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \bar{X}_n)^2 \text{ (unbiased, slightly higher var)}$$

vs. $\frac{1}{n}\sum(X_i - \bar{X}_n)^2$ (biased but lower MSE for some n). Choice depends on which loss / risk you care about.

Estimation ↔ ERM

ERM perspective

- Learning setup: data $Z = (X, Y) \sim F$; *risk* $R(g) = \mathbb{E}[L(Z, g)]$ and *empirical risk* $\hat{R}_n(g) = \frac{1}{n}\sum_{i=1}^{n} L(Z_i, g)$. (Ch. 4)
- *ERM*: $\hat{g} \in \arg\min_{g\in\mathcal{M}} \hat{R}_n(g)$. With log-loss $L(z, \alpha) = -\log p_\alpha(z)$, ERM coincides with *MLE*.

Risk decomposition Definitions

- $g^* = \arg\min_g R(g)$: For instance, the Bayes rule (best possible predictor, may be unattainable).
- $g^*_{\mathcal{M}} = \arg\min_{g\in\mathcal{M}} R(g)$: best predictor within model $\mathcal{M}$.
- $\hat{g} \in \arg\min_{g\in\mathcal{M}} \hat{R}_n(g)$: ERM predictor learned from the data.
- True risk is decomposed as *Approximation* + *estimation* decomposition: $R(\hat{g}) - R(g^*) = \underbrace{R(g^*_{\mathcal{M}}) - R(g^*)}_{\text{approximation error}} + \underbrace{R(\hat{g}) - R(g^*_{\mathcal{M}})}_{\text{estimation error}}$, where $g^*_{\mathcal{M}} = \arg\min_{g\in\mathcal{M}} R(g)$. (Ch. 4)

From estimation to generalization I

Training vs testing viewpoint

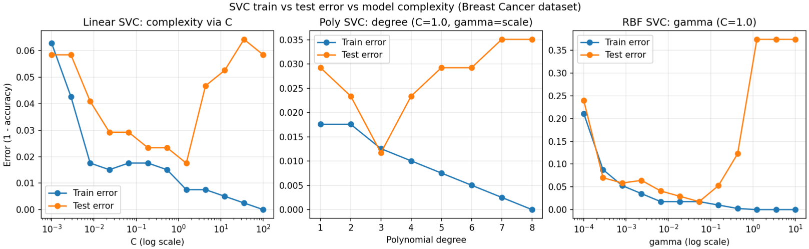
Setup

- **Training data** $\{Z_i = (X_i, Y_i)\}_{i=1}^n$: used to fit $\hat{g}$ by minimizing empirical risk $\hat{R}_{\text{train}}(g) = \frac{1}{n}\sum_{i=1}^n L(Z_i, g)$.
- **Test data** $\{Z_j^{\text{test}}\}_{j=1}^m$: never seen during training; used to estimate the *true* (population) risk $R(g) = \mathbb{E}[L(Z, g)]$ via $\hat{R}_{\text{test}}(g) = \frac{1}{m}\sum_{j=1}^m L(Z_j^{\text{test}}, g)$.

Train / test performance gap

- On training data, ERM picks $\hat{g} \in \arg\min_{g\in\mathcal{M}} \hat{R}_{\text{train}}(g)$. Good *training error* does not automatically mean good *test error*.
- **Overfitting**: $\hat{R}_{\text{train}}(\hat{g})$ very small but $\hat{R}_{\text{test}}(\hat{g})$ large (model fits noise).
- **Underfitting**: both train and test errors large (model too simple).

Train vs test illustration



- As model complexity increases, training error decreases (more flexible model).
- Test error initially decreases (better fit) but eventually increases (overfitting).
- Estimation of more parameters with limited data increases variance, harming generalization.
- Goal: find model complexity that minimizes test error.

From estimation to generalization II

Concentration ⇒ reliable test evaluation

- For bounded losses $L \in [0,1]$, on i.i.d. test data of size m:

$$\Pr\big(|\hat{R}_{\text{test}}(g) - R(g)| > \varepsilon\big) \leq 2e^{-2m\varepsilon^2} \quad \text{(Hoeffding).}$$

  So a large independent test set makes $\hat{R}_{\text{test}}(g)$ a sharp estimate of the true risk.

- For multiple candidate models $g \in \mathcal{M}$ evaluated on the same test set, union bound gives

$$\Pr\Big(\sup_{g \in \mathcal{M}} |\hat{R}_{\text{test}}(g) - R(g)| > \varepsilon\Big) \ \leq\ 2|\mathcal{M}|\, e^{-2m\varepsilon^2}.$$

  This quantifies how reliable model comparison on a finite test set is.

The corresponding CIs scale as $O\big(\sqrt{\frac{\log|\mathcal{M}|}{m}}\big)$.

Ch.6 RNG: LCG, period and basic sampling

Linear Congruential Generator (LCG)

$$u_{k+1} = (au_k + c) \bmod M$$

Full period M if

1. $\gcd(c, M) = 1$,
2. for every prime $p \mid M$: $p \mid (a-1)$,
3. if $4 \mid M$: $4 \mid (a-1)$.

Scaling $u_k/M \in [0,1]$ approximates Uniform; quality checked via moments and correlations. **Basic sampling**

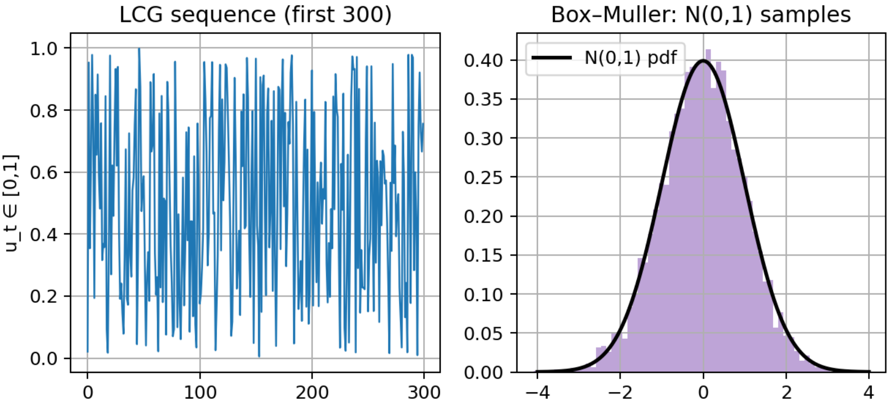
- **Inversion:** $X = F^{-1}(U)$ works for any CDF when F^{-1} is tractable.
- **Accept-Reject:** draw $Y \sim g$, $U \sim \text{Unif}[0,1]$; accept Y if $U \leq f(Y)/(Mg(Y))$. Efficiency $\approx 1/M$ where $M \geq \sup_x f(x)/g(x)$.

Ch.6 RNG: Box-Muller for Gaussians

Goal: sample $Z_0, Z_1 \sim \mathcal{N}(0,1)$ i.i.d. from $U_1, U_2 \sim \text{Unif}[0,1]$. **Box-Muller transform**

$$Z_0 = \sqrt{-2\log U_1}\cos(2\pi U_2), \qquad Z_1 = \sqrt{-2\log U_1}\sin(2\pi U_2).$$

Uses polar coordinates and the joint density of independent standard Gaussians.



Ch.7 Markov chains: definition

Definition

- A homogeneous **Markov chain** $(X_t)_{t\geq 0}$ on state space $\mathcal{X}$ is a sequence of random variables satisfying the *memoryless property*:

$$\mathbb{P}(X_{t+1} = y \mid X_t = x, X_{t-1}, \ldots, X_0) = \mathbb{P}(X_{t+1} = y \mid X_t = x) = P(x, y).$$

- The *transition matrix* $P = (P(x, y))_{x,y\in\mathcal{X}}$ is row-stochastic: $P(x, y) \geq 0$ and $\sum_y P(x, y) = 1$.
- Evolution of the distribution: $p_{t+1} = p_t P$; by induction $p_t = p_0 P^t$.

Random Mapping Representation (RMR)

- Simulate via $X_{t+1} = \rho_t(X_t, W_t)$ with $\mathbb{P}(\rho_t(x, W) = y) = P(x, y)$, where (W_t) are i.i.d. random variables.

Ch.7 Markov chains: structure and convergence

Key properties

- **Irreducible**: from any state x we can reach any state y with positive probability in some number of steps.
- **Aperiodic**: the gcd of return times to each state is 1 (rules out deterministic cycles).
- **Stationary distribution** π: a probability vector with $\pi P = \pi$. If the chain is finite, irreducible and aperiodic, then

  $$p_t \xrightarrow[t\to\infty]{} \pi$$

  for any initial distribution p_0.
- **Reversible** (detailed balance): $\pi(x)P(x,y) = \pi(y)P(y,x)$ for all $x, y \Rightarrow \pi$ is stationary.

Ch.8 Large-margin classifiers: hard and soft margin

Setup: binary classification with large margin

- Data: (x_i, y_i) with $x_i \in \mathbb{R}^d$, $y_i \in \{-1, +1\}$.
- Hyperplane: $\{x : w^\top x + b = 0\}$ with normal w.
- Margin: geometric distance from hyperplane to nearest point.

Goal: find a hyperplane with *maximum margin* between two classes.

- Hard-margin SVM (separable case):

$$\min_{w,b} \ \tfrac{1}{2}\|w\|^2 \quad \text{s.t.} \quad y_i(w^\top x_i + b) \geq 1 \ \forall i.$$

  Maximizing geometric margin $= 1/\|w\|$.

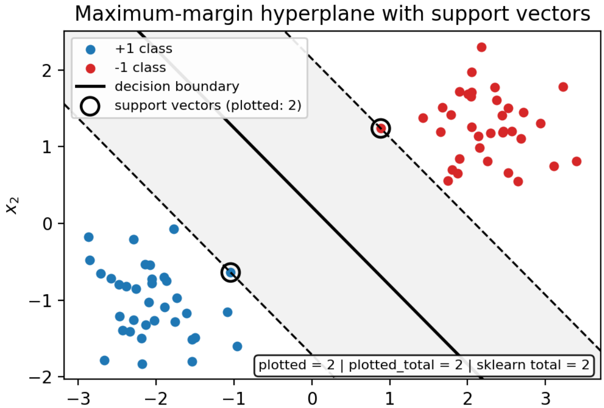
- Soft-margin SVM (hinge loss):

$$\min_{w,b} \ \frac{1}{n}\sum_{i=1}^{n} \max(0, 1 - y_i(w^\top x_i + b)) + \frac{\lambda}{2}\|w\|^2.$$

Large margin improves robustness.

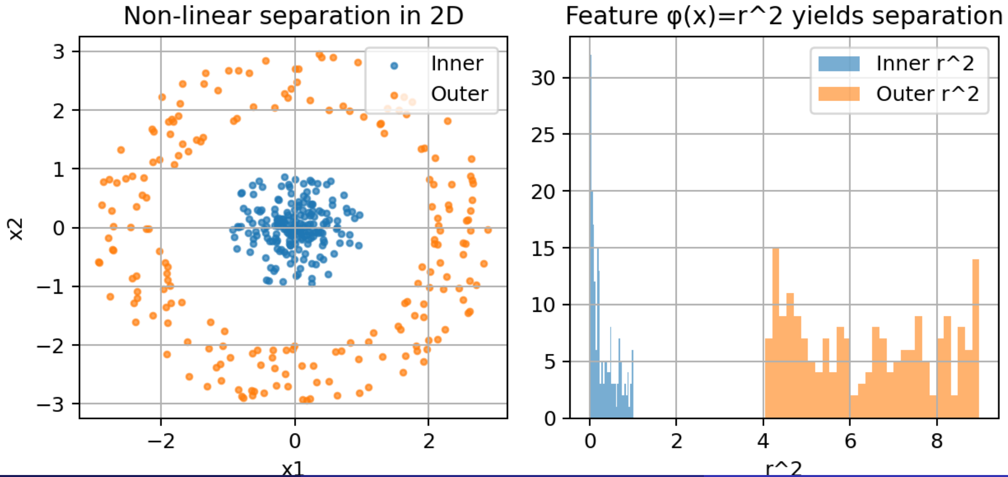
Ch.8 Large-margin classifiers: support vectors

- Only **support vectors** (points with $y_i(w^\top x_i + b) \leq 1$) determine w.
- Non-support vectors could be removed without changing the classifier.
- Kernel trick turns this into nonlinear SVM: replace $x_i^\top x_j$ by $k(x_i, x_j)$.



Ch.8 Kernelization and testing

Kernel trick: choose PSD (positive semidefinite) kernel $k(x, y) = \phi(x)^\top \phi(y)$; perceptron uses $w = \sum_i c_i \phi(x_i)$; decision by $\sum_i c_i k(x_i, x)$. Common kernels: linear, polynomial, RBF.



Ch.8 Kernel trick: weight and loss formulation

Dual representation via kernels

- Feature map: $\phi(x) \in \mathcal{H}$ (possibly infinite-dimensional Hilbert space).
- Weight vector: $w = \sum_{i=1}^{n} c_i \phi(x_i)$ for coefficients $c_i \in \mathbb{R}$.
- Prediction: $\hat{y}(x) = \operatorname{sign}\left(\sum_{i=1}^{n} c_i k(x_i, x) + b\right)$ where $k(x_i, x) = \phi(x_i)^\top \phi(x)$ is the kernel.

Soft-margin SVM loss in dual form

- Original primal (hinge loss):

$$\min_{w,b} \frac{1}{n} \sum_{i=1}^{n} \max(0, 1 - y_i(w^\top \phi(x_i) + b)) + \frac{\lambda}{2} \|w\|^2.$$

- Substituting $w = \sum_j c_j \phi(x_j)$:

$$\min_{c,b} \frac{1}{n} \sum_{i=1}^{n} \max(0, 1 - y_i(\sum_j c_j k(x_i, x_j) + b)) + \frac{\lambda}{2} \sum_{i,j} c_i c_j k(x_i, x_j).$$

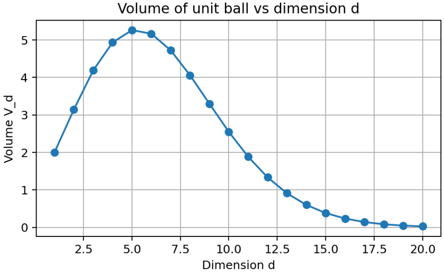
This depends on x_i, x_j only through $k(x_i, x_j)$, avoiding explicit ϕ computation.

Ch.9 High-dimensional geometry: volumes

- Unit-ball volume: $V_d = \dfrac{\pi^{d/2}}{\Gamma(\frac{d}{2}+1)}$. Using Stirling for large d:

$$V_d \approx \frac{1}{\sqrt{\pi d}} \left(\frac{2\pi e}{d}\right)^{d/2},$$

  which decays super-exponentially once d is moderately large.
- Sampling: normalize Gaussian to sample on sphere; multiply by $U^{1/d}$ for ball.



Ch.9 Annulus effect in high dimensions

Annulus effect via volumes

- Consider the outer shell

$$A_\epsilon = B_1 \setminus B_{1-\epsilon} = \{x : 1-\epsilon \le \|x\| \le 1\}.$$

- Using $|B_r| = r^d |B_1|$, its relative volume is

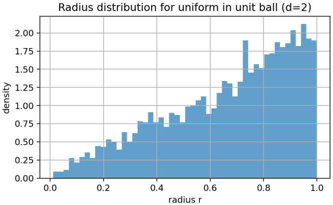
$$\frac{|A_\epsilon|}{|B_1|} = \frac{|B_1| - |B_{1-\epsilon}|}{|B_1|} = 1 - (1-\epsilon)^d.$$

- For fixed $\epsilon > 0$, this ratio $\to 1$ as $d \uparrow \infty$: almost all volume lies in a thin outer annulus.

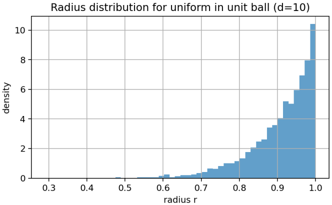
Ch.9 Radius distribution and annulus visualization

Radius distribution in high dimensions

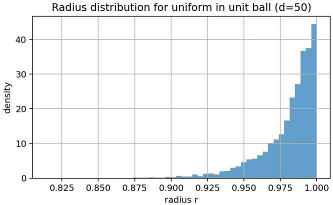
- Let X be uniformly distributed in the unit ball of $\mathbb{R}^d$ and $R = \|X\|$ its radius.
- As d increases, most of the probability mass for R concentrates near the boundary $R \approx 1$.



$d = 2$



$d = 10$



$d = 50$

Ch.10 Johnson-Lindenstrauss (JL) lemma

For n points, with $k = O(\epsilon^{-2} \log n)$ there exists a linear map $R : \mathbb{R}^D \to \mathbb{R}^k$ such that

$$(1-\epsilon)\|x_i - x_j\| \le \|Rx_i - Rx_j\| \le (1+\epsilon)\|x_i - x_j\| \quad \forall i < j.$$

Relative distortion for a pair (i,j):

$$\delta_{ij} \;=\; \frac{d_k - d_h}{d_h},$$

where $d_h = \|x_i - x_j\|$ is the high-dimensional distance and $d_k = \|Rx_i - Rx_j\|$ is the distance after projection.

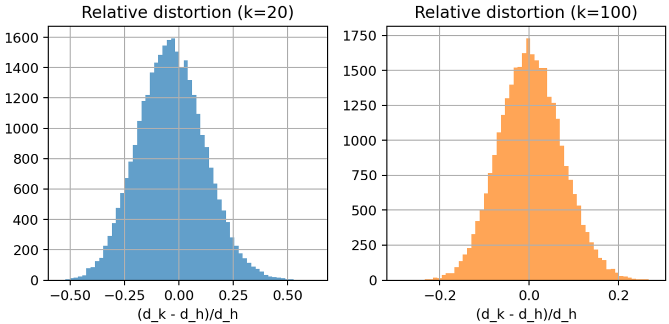
- $\delta_{ij} = 0$: exact preservation of that distance.
- $\delta_{ij} > 0$: distance slightly stretched.
- $\delta_{ij} < 0$: distance slightly shrunk.

Ch.10 JL: distortion distributions for $k = 20$ vs $k = 100$

- For $k = 20$ (red curve in the figure), the distribution of δ_{ij} is wider: some pairs get more distortion (both positive and negative).
- For $k = 100$ (blue curve), the distribution of δ_{ij} is narrower and more concentrated around 0:

$$\|Rx_i - Rx_j\| \approx \|x_i - x_j\| \quad \text{for most pairs.}$$

- Increasing $k \Rightarrow$ better distance preservation, but higher computational/storage cost.



Ch.10 SVD/PCA: matrix as data cloud

Data matrix as points in $\mathbb{R}^d$

- Let $A \in \mathbb{R}^{n\times d}$, with rows $a_i^\top$. After centering columns, each row $a_i^\top$ is a data point in $\mathbb{R}^d$.
- We study the scatter matrix

  $$S = A^\top A$$

  whose eigenvalues/eigenvectors describe how variance is distributed across directions.
- A direction $u \in \mathbb{R}^d$ has variance

  $$\mathrm{Var}(Au) = u^\top S u.$$

  Maximizing this over $\|u\| = 1$ gives the first principal component.

Goal of PCA

- Find an orthonormal basis $u_1, \ldots, u_d$ such that the projected data Au_k has decreasing variance:

  $$\mathrm{Var}(Au_1) \geq \mathrm{Var}(Au_2) \geq \cdots \geq 0.$$

- The low-dimensional representation keeps only the first k coordinates in this basis.

Ch.10 SVD/PCA: decomposition and variance explained

SVD of the centered data matrix

- Singular Value Decomposition:

$$A = U \Sigma V^\top$$

  where $U \in \mathbb{R}^{n \times r}$, $V \in \mathbb{R}^{d \times r}$ have orthonormal columns, $\Sigma = \mathrm{diag}(\sigma_1, \dots, \sigma_r)$ with $\sigma_1 \geq \cdots \geq \sigma_r > 0$, and $r = \mathrm{rank}(A)$.
- **Right singular vectors** (columns of V): principal directions in feature space (eigenvectors of $A^\top A$).
- **Left singular vectors** (columns of U): directions in the sample space (eigenvectors of $AA^\top$).

PCA as SVD + low-rank approximation

- Best rank-k approximation (Eckart–Young):

$$A_k = U_k \Sigma_k V_k^\top,$$

  where U_k, Σ_k, V_k keep only the first k singular values/vectors.
- Low-dimensional coordinates of data points: rows of $U_k \Sigma_k$ (or equivalently AV_k) are the k-dimensional embeddings of the original points.

Wrap-up: key take-aways

- **Risk & estimation:** the risk view unifies classical estimation and machine learning; log-loss makes ERM $=$ MLE in many models.
- **Generalization:** concentration inequalities (Hoeffding, DKW, union bound) explain why held-out / test performance can reliably estimate true risk.
- **Randomness & dynamics:** RNG methods (LCG, inversion, accept-reject, Box-Muller) and Markov chains let us simulate complex systems and study their steady-state behaviour.
- **Learning in high dimensions:** large-margin methods (perceptron, SVM, kernels), high-dimensional geometry, JL, and SVD/PCA are core tools for classification and dimensionality reduction.