CLASSIFIERS

$\{x : f(x) = 0\}$; isosurface for 0

Decision boundary: $(d-1)$ dim surface

linear C: decision b. line/plane

LINALG

unit vec: $\frac{x}{\|x\|}$

$f(x) = w \cdot x + \alpha$, $H = \{w^T x = -\alpha\}$

$\vec{w} \perp H$, $w^T(y - x) = 0$, y, x on H

$\vec{w}$ unit vector $\Rightarrow f(x)$ signed dist from x to H

α: Distance from origin

CENTROID METHOD: mean of μ_c, $\mu_{\text{not-}c}$

↳ $f(x) = (\mu_c - \mu_{\neg c}) \cdot x - (\mu_c - \mu_{\neg c}) \cdot \frac{\mu_c + \mu_{\neg c}}{2}$

PERCEPTRON METHOD: for linearly separable

↳ Uses gradient descent — $O(r^2/\gamma^2)$ iter (↳ max rad, max margin)

↳ $y_i = \begin{cases} 1 & \text{in } c \\ -1 & \text{not in } c \end{cases}$

↳ $X_i \cdot w \geq 0$, $y_i = 1$

$X_i \cdot w \leq 0$, $y_i = -1$

ADD 1 bias term

$y_i X_i \cdot w \geq 0$

LOSS FUNCTION: $L(z, y_i) = \begin{cases} 0 & , y_i z \geq 0 \\ -y_i z & , \text{otw} \end{cases}$

RISK: $\frac{1}{n} \sum L(X_i \cdot w, y_i) = \frac{1}{n} \sum -y X_i w$

$\nabla R(w) = -\sum y_i X_i$

while some $y_i X_i \cdot w < 0$, $w^t \leftarrow w^{t-1} + \sum y_i X_i$

Fictitious dim: Add α to w, 1 to X

SVM

margin: dist from from decision b to nearest training point

dist from X_i: $\frac{w \cdot X_i}{\|w\|} + \frac{\alpha}{\|w\|}$

margin: $\min \frac{1}{\|w\|} |w \cdot X_i + \alpha| \geq \frac{1}{\|w\|}$

HARD MARGIN:

$\min \|w\|^2$ (smooth func)

s.t. $y_i (X_i \cdot w + \alpha) \geq 1$

Optimal: margin $= \frac{1}{\|w\|}$

Slab: $\frac{2}{\|w\|}$

linearly separable, outliers ∴

SOFT MARGIN:

$\min \|w\|^2 + C \sum \xi_i$

s.t. $y_i (X_i \cdot w + \alpha) \geq 1 - \xi_i$

$\xi_i \geq 0$

small C: max margin $\frac{1}{\|w\|}$, min $\|w\|$, underfit

less sensitive, more flat

big C: $\xi_i \to 0$, small margin, overfit

very sensitive, approach hard margin

Parabolic Lifting Map: $\Phi(x) = \begin{bmatrix} x \\ \|x\|^2 \end{bmatrix}$

Ellipsoid: axis aligned, no cross terms

$Ax_1^2 + Bx_2^2 + Cx_3^2 + Dx_1x_2 + \ldots + Ix_3 + \alpha = 0$

$f(x) = [A \; B \; C \; \cdots \; I] \cdot \Phi(x) + \alpha$

quadric ⇔ conic in 2D

overfit if too high ∴

Raising: linearly separable ✓ wider boundary robust

Edge Detection: Find regions with high gradients

OPTIMIZATION

↳ convex func: no min, 1 min, ∞ min

↳ strictly convex: 1 global min

↳ high ellipticity of contours ($\lambda_{max} \ggg \lambda_{min}$), no good learning rate

↳ convex polytope satisfies linear program (if non empty, linearly separable)

DECISION THEORY

$P(A \cap B) = P(A) P(B|A)$

$P(A|B) = \frac{P(A \cap B)}{P(B)}$

Bayes Theorem:

$P(Y=1|X) = \frac{P(X|Y=1) P(Y=1)}{P(X)}$ (posterior)

$R(r) = E[L(r(X), Y)]$

$= \sum_x [L(r(x), 1) P(Y=1|X) + L(r(x), -1) P(Y=-1|X)] P(X)$

$r^*(x) = \begin{cases} 1 \\ -1 \end{cases}$ if $L(-1,1) P(Y=1|X) > L(1,-1) P(Y=-1|X)$ otw

When L symmetric, pick class with biggest posterior

$E[g(X)] = \int_{-\infty}^{\infty} g(x) f(x) \, dx$ $\quad \text{Var}(X) = E[(X - E[X])^2]$

$\mu = E[X] = \int_{-\infty}^{\infty} x f(x) \, dx$ $\quad \sigma^2 = E[(X-\mu)^2] = E[X^2] - \mu^2$

Bayes Decision Boundary:

$P(X|Y=1) P(Y=1) = P(X|Y=0) P(Y=0)$

Bayes Risk: Area under min of functions

GAUSSIAN DIS. ANALYSIS

$X \sim N(\mu, \sigma^2)$: $f(x) = \frac{1}{(\sqrt{2\pi}\sigma)^d} \exp\left(\frac{\|x-\mu\|^2}{-2\sigma^2}\right)$ concave

Bayes maximises $f(X|Y=c) \pi_c$

Equivalently, $Q_c(x) = -\frac{\|x-\mu\|^2}{2\sigma^2} - d \ln c + \ln \pi_c$

Models $P(Y=y|x)$ as a logistic $s = \frac{1}{1+e^{-\gamma}}$

QDA

SAMPLE variances

↳ Pick class to maximise Q_c

↳ 2 class: $Q_c(x) - Q_D(x) = 0$ (DB)

$P(Y=c|X) = \frac{f_{X|Y=c} \, \pi_c}{f_{X|Y=c} \pi_c + f_{X|Y=D} \pi_D} = s(Q_c - Q_D)$

LDA

All gaussians have same variance

$\max \frac{\mu_c \cdot x}{\sigma^2} - \frac{\|\mu_c\|^2}{2\sigma^2} + \ln \pi_c$

2 class: DB: $w \cdot x + \alpha = 0$, $P(Y=c|X) = s(DB)$

MLE

$\hat{\mu} = \frac{1}{n} \sum X_i$ $\quad \hat{\sigma}_2 = \frac{1}{n} \sum \|X_i - \hat{\mu}\|^2$

LDA: Same means and priors, one variance

$\hat{\sigma}_2 = \frac{1}{n} \sum_c \sum \|X_i - \hat{\mu}_c\|^2$

QDA: Same mean $\hat{\mu}$, calc $\hat{\sigma}_c^2$, $\hat{\pi}_c = \frac{n_c}{n}$

$d(\mu, \sigma : X_1, \ldots X_n) = f(X_1) f(X_2) \cdots f(X_n)^{\sum n_D}$

$\ell(\ldots) = \ln f(X_1) + \ldots \ln f(X_n)$

$= \sum \left(-\frac{\|X_i - \mu\|^2}{2\sigma^2} - d \ln \sqrt{2\pi} - d \ln \sigma \right)$

EIG

$\vec{v}_i, \lambda_i$ for $A \to \vec{v}_i, \lambda_i^k$ for A^k

$\to \vec{v}_i, \frac{1}{\lambda_i}$ for A^{-1}

Spectral Theorem: Real, symmetric M has real λ_i, n eigenvectors which are mutually orthogonal

$M = V \Lambda V^T$ (UNIT eigenvectors)

PSD:

$x^T A x \geq 0$, $\lambda_i \geq 0$, $A = UU^T$

$A^{1/2}$ exists

VISUAL

$x^T A^{-1} x = \|A^{-1/2} x\|_2^2$ is an ellipsoid

axes $\vec{v}_i \{A\}$, radii $\lambda_i \{A\}$

A diagonal: axis aligned

MULTIV. GAUSSIAN

$X \sim N(\mu, \Sigma)$

$f(x) = \frac{1}{\sqrt{(2\pi)^d |\Sigma|}} \exp\left(-\frac{1}{2}(x-\mu)^T \Sigma^{-1} (x-\mu)\right)$

Σ: covariance, Σ^{-1}: precision

Isocontours of $(x-\mu)^T \Sigma^{-1} (x-\mu)$ determined by eigen pairs of $\Sigma^{1/2}$

Covariance

$\text{Cov}(R, S) = E[(R - E[R])(S - E[S])^T]$

$= E[RS^T] - \mu_R \mu_S^T$

Var(R): PSD matrix

R_i, R_j ind $\Rightarrow \text{Cov}(R_i, R_j) = 0$

$\text{Cov}(R_i, R_j) = 0$ and multivariate norm $\Rightarrow R_i, R_j$ independent

All features ind $\Rightarrow$ Var(R) diagonal

Var(R) diagonal + joint normal ⇔ $f(x) = f(x_1) \cdots f(x_n)$

⇔ ellipsoids axis aligned, squared radii on Σ diagonal

ANISOTROPIC

LDA: $\frac{1}{n} \sum_c \sum_{y_i = c} (X - \hat{\mu}_c)(X_i - \hat{\mu}_c)^T$

QDA: $\frac{1}{n_c} \sum_{y_i = c} (X - \hat{\mu}_c)(X_i - \hat{\mu}_c)^T$

↳ $Q_c - Q_D$ is quadratic

↳ DB quadric

$-\frac{1}{2}(x-\mu_c)^T \Sigma_c^{-1} (x-\mu_c) - \frac{1}{2} \ln |\Sigma_c| + \ln \pi_c$

Multi-LDA:

$\max \mu_c^T \Sigma^{-1} x + \frac{\mu_c^T \Sigma^{-1} \mu_c}{2} + \ln \pi_c$

LDA underfits, QDA overfits

$\dot{X} = X - \mu$

$\text{Var}(X) = \frac{1}{n} \dot{X}^T \dot{X}$

$Z = V^{-1} X$

Decorrelate X: $Z = XV$, $\text{Var}(R) = V \Lambda V^T$

$X \sim N(\mu, \Sigma)$ $Z \sim N(0, \Lambda)$

Whitening: center + sphering
Sphering X: $W = X\ \mathrm{Var}\ R^{-1/2}$

REGRESSION

1. Linear: $w \cdot x + \alpha$
2. poly
3. logistic: $s(w \cdot x + \alpha)$

Loss functions

A. $(z-y)^2$ squared
B. $|z-y|$ absolute
C. $-y \ln z - (1-y)\ln(1-z)$ logistic (0-1 labels)

Cost functions

a. $\frac{1}{n}\sum L(h(x_i), y_i)$ mean
b. $\max L(h(x_i), y_i)$ max
c. $\sum w_i L(h(x_i), y_i)$ weighted
d. $a/b/c + \lambda\|w\|^2$ l_2
e. $a/b/c + \lambda\|w\|_1$ l_1

LEAST SQUARES

$\min \sum_i (x_i \cdot w + \alpha - y_i)^2$

$\min \|Xw - y\|^2$

$\nabla RSS: X^T X w = X^T y$. $X^T X$ always PSD

$w = \underbrace{(X^T X)^{-1} X^T}_{\text{pseudoinverse}}\, y$

$\hat{y} = Xw = XX^+ y = Hy$. Ideal $H = I$

Unique when $X^T X$ singular ✓
Sensitive to outliers ✗

LOGISTIC REGRESSION

cost func CONVEX

$J = \sum_i L(s(x_i \cdot w), y_i)$

LINEAR db for linear input

$= -\sum_i (y_i \ln s_i + (1-y_i)\ln(1-s_i))$

$s'(\gamma) = s(\gamma)(1-s(\gamma))$

$\nabla_w J = -X^T(y - s)$

$\nabla_w^2 J = X^T \Omega X$ PSD $\quad \Omega = \mathrm{diag}(s_i(1-s_i))$

BGD: $w \leftarrow w + \epsilon X^T(y - s(Xw))$

SGD: $w \leftarrow w + \epsilon (y_i - s(x_i \cdot w)) x_i$

separates linearly separable points

WEIGHTED LEAST SQUARES

$\min (Xw - y)^T \Omega (Xw - y)$

$\Rightarrow X^T \Omega X w = X^T \Omega y$

NEWTON'S METHOD

Iterative optimisation method for smooth $J(w)$

Works when $J(w)$ is quadratic

$w \leftarrow w - (\nabla^2 J)^{-1}(\nabla J)$

$w \leftarrow w + e, \quad (X^T \Omega X)e = X^T(y - s)$

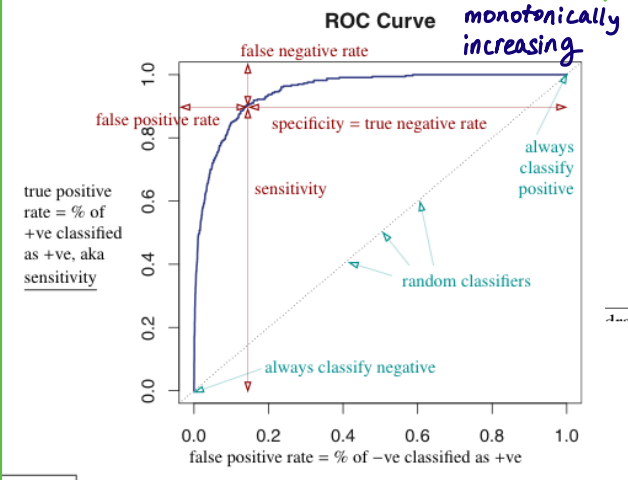
small weight in W ⇒ more emphasis
faster than SGD, esp when d small
NOT on perceptron

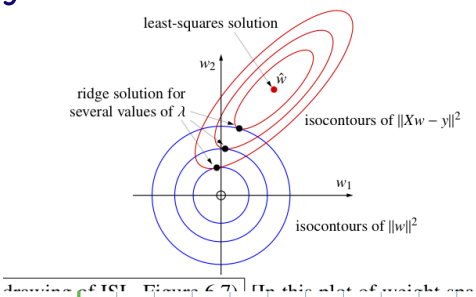
$\nabla_x(\alpha \vec{y}) = (\nabla_x \alpha)\vec{y}^T + \alpha \nabla_x \vec{y}$

$\nabla_x \vec{f}(\vec{y}) = (\nabla_x \vec{y})(\nabla_{\vec{y}} \vec{f}(\vec{y}))$

$\nabla_x(\vec{y} \cdot \vec{z}) = (\nabla_x y)z + (\nabla_x z)y$

$\nabla_x g(\vec{y}) = \nabla_x(y)\, \nabla_y(g(y))$





drawing of ISL Figure 6.7) [In this plot of weight space

Knobs: posterior prob threshold for GDA/Log Reg
Based on real test data

Bayes classifier has nothing to do w sample points, needs to know abt distribution

Ellipse eqn requires different quad coeffs

$P(x \geq t) \leq \frac{E[x]}{t}$

$\Sigma = E[(z-\mu)(z-\mu)^T]$

$P(A \cap B) = P(A|B)\,P(B)$

$R(r(x) = i|x) = \sum L(r(x) = i, y = j)\, P(Y = j|x)$

bias $= E[\hat{\theta}] - \theta$

BIAS-VARIANCE

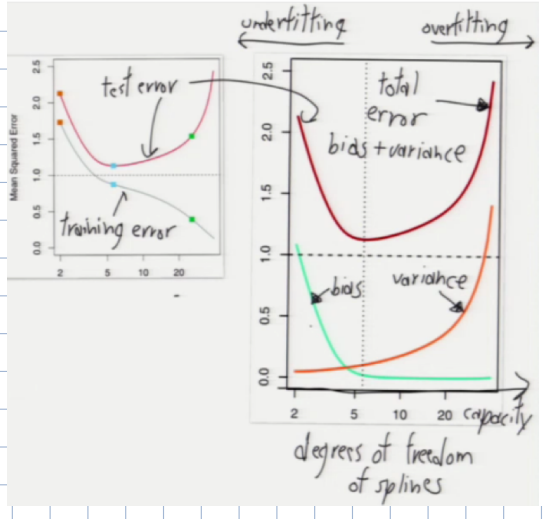
Bias: Inability of h fitting g
Variance: Random noise in data
Overfitting: Bias ↓ Variance ↑
Underfitting: Bias ↑ Variance ↓
+ Feature: Variance ↑ Bias =, ↓

$E[L(h(z)), y] = E[(h(z) - y)^2]$

$= E[h(z)^2] + E[y^2] - 2E[yh(z)]$

$= \mathrm{Var}(h(z)) + E[h(z)]^2 + \mathrm{Var}(\delta) + E[\delta]^2 - 2E[yh(z)]$

$= \underbrace{(E[h(z)] - g(z))^2}_{\text{bias}^2} + \underbrace{\mathrm{Var}(h(z))}_{\text{var}} + \underbrace{\mathrm{Var}(\epsilon)}_{\text{noise}}$



variance more harmful to test error
hyperparameters (λ, C, α) affect b-v

$w^* = X^+ y$ (least squares soln. of LEAST NORM)

LDA convex polytope (S subset of union of hyperplanes, S connected)

$\mathrm{rank}(X) + \dim(N(A)) = n$

$\dim(\mathrm{row}(X)) = \dim(\mathrm{col}(X^T))$
$= \mathrm{rank}(X^T)$
$= \mathrm{rank}(X)$

transforming data changes scale
shifting does nothing

$\langle s \cdot t \rangle \leq \|s\|\|t\|$

$A \Rightarrow B,\ B \Rightarrow A,\ \neg A \Rightarrow \neg B$

Σ diag ⇒ diag $(\sigma_1^2, \sigma_2^2, \ldots \sigma_n^2)$

RIDGE

$\min \|Xw - y\|^2 + \lambda\|w'\|^2$

ALWAYS unique optimum, ONLY one

$\lambda \to \infty$, var → 0, bias ↑, $w^* \to 0$

$\underbrace{(X^T X + \lambda I)}_{PD}\, w = X^T y$

Bayesian Justification: $w' \sim N(0, c^2)$

$f(w') \propto e^{-\|w'\|^2/2c^2}$

$\max \ln f_{y|x} + \ln f(w') - \text{const}$
$= -\text{const}\,\|Xw - y\|^2 - \text{const}\,\|w'\|^2$
$= \min \|Xw - y\|^2 + \lambda\|w'\|^2$

Gaussian w $\Sigma = cI$ prior

LASSO

$\min \|Xw - y\|^2 + \lambda\|w'\|_1$ quadratic prog

sparsity (intersects w axes)

FEATURE

feature ↑, variance ↑

FORWARD: start with 0 features, keep adding until Val error ↑
few ☺ won't find best 2-f model

BACKWARD: start with d, remove if removing ↓ val error
more ☺

Train data ↓, Train Accuracy ↑

DECISION TREES

↳ classification + regression
↳ Tree with 2 nodes (internal nodes / feature + leaf)

$P_c = \frac{\#\text{ points in } C}{\#\text{ total pts}}$

Entropy: $H(S) = -\sum_c p_c \log_2 p_c$ (all same = 0, half C, half D = 1, n dif classes = $\log_2 n$)

Weighted Avg. Entropy: $H_a = \frac{|S_l| H(S_l) + |S_r| H(S_r)}{|S_l| + |S_r|}$

Info gain: $H(S) - H_a$ — Choose split to MAX info gain, ALWAYS non-negative (concave)

Test: $\le O(d)$, usually $\le O(\log N)$

Train: $\le O(nd)$

↑ fast, bias ↓, translation invariant, robust to irrelevant features
↑ variance

VARIATIONS

Regression: Average sample points at each node
↳ Piecewise constant regression fn (Label = μ_S, Cost = Var_S)
↳ choose split to MIN weighted avg. variance

Stopping Early: → Limits tree depth, tree size (speed, big data)
• Stops pure tree overfitting, good for overlapping distbns.
• Return posterior prob. at each leaf (classification (majority) / regression (average))

Pruning: Grow big tree, kill split if validation ↓
↳ More reliable than stopping early

Multivariate Split: Non-axis aligned splits w other algs. Classification ↑ Building Speed ↓

Ensemble Learning: Reduces variance in DTs
↳ **Bagging**: Same learning alg, on random subsamples (WITH replacement), T learners
↳ **Random Forests**: Bagging + feature subset m
$m \downarrow \Rightarrow$ randomness ↑, bias ↑, tree correlation ↓

KERNEL

$\Phi: \mathbb{R}^d \to \mathbb{R}^D$. d input features, D after featurization

Degree p-polynomials blow upto $D \in O(d^p)$ features

In many algs, $w = X^T a = \sum_n a_i x_i$ (linear combo of pts) and can use inner product of $\phi(x)$ only.

Optimize n dual weights $\vec{a}$ instead of D primal $\vec{w}$

Solve $n \times n$ linear system instead of $d \times d$

Kernel Ridge Regression:

$(X^TX + \lambda I) w = X^T y$

$(XX^T + \lambda I) a = y$ [works if you center X and y so $E[\text{bias}] \approx 0$]

Then $X^T y = (X^TX + \lambda I) X^T a$

$\Rightarrow w = X^T a$ is a solution.

Obj: $\min_a \|X^TXa - y\|^2 + \lambda \|X^Ta\|^2$

Train: Solve $(XX^T + \lambda I) a = y$

Test: $h(z) = w^T z = a^T X z = \sum_n a_i (X_i^T z)$

Kernel fn: $k(x_1, x_2) = x_1^T x_2$. $K = XX^T$ ($n \times n$)

$K_{ij} = k(X_i, X_j)$ (Use dual when $d \gg n$)

Alg: $K_{ij} \leftarrow k(X_i, X_j)$ $O(n^2 d)$
Solve $(K + \lambda I) a = y$ $O(n^3)$
Test: $h(z) = \sum_n a_i k(X_i, z)$ $O(nd)$

Polynomial Kernel: Degree p is $k(x_1, x_2) = (x_1^T x_2 + 1)^p = \phi(x_1)^T \phi(x_2)$

Compute $\phi(x_1)^T \phi(x_2)$ in $O(d)$ instead of $O(d^p)$

SGD dual: $a_i \leftarrow a_i + \epsilon (y_i - s(Ka)_i)$

Gaussian Kernel: generate feature vecs in ∞ space

$\phi: \mathbb{R}^d \to \mathbb{R}^\infty$. $k(x_1, x_2) = \exp\left(-\frac{\|x_1 - x_2\|^2}{2\sigma^2}\right)$ $O(d)$

inner products of similar features large, small for dissimilar features

Key: $h(z) = \sum_n a_j k(X_j, z)$ is linear combo of Gaussians centered at training pts

Dual weights coefficients of linear combo

Why?: Very smooth h, like K-NN but smoother, oscillates less than polynomials

larger σ → wider gaussian → smoother h → bias ↑↑ variance ↓↓

NEURAL NETWORKS

↳ classification + regression
↳ perceptrons failed @ XOR
↳ non-linear activation $s'(z) = s(z)(1 - s(z))$

1 hidden layer:

Input: $x_1, \dots x_d$ Layer 1 weights: $m \times (d+1)$
Hidden: $h_1, \dots h_m$ Layer 2 weights: $K \times (m+1)$
Output: $z_1 \dots z_K$

Training: SGD / Batch. Cost function NOT convex

Initialize random weights to break symmetry, $\mu = 0$, $\sigma = \frac{1}{\sqrt{n}}$

Backprop: $O(\text{edges})$, dp alg

Vanishing Gradient: s close to 0 or 1, $s' \to 0$
Gradients change too slowly. Layers ↑ Vanishing ↑
Sigmoid → ReLU: $\max\{0, y\}$, might explode

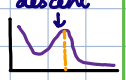
Output Unit

① Regression: linear output units + squared loss
② Classification: classes ≥ 2, softmax units $z_i(t) = \frac{e^{t_i}}{\sum_{j=1}^K e^{t_j}}$, cross entropy $-\sum_{i=1}^K y_i \ln z_i$

linear / ReLU → squared error
sigmoid / softmax → cross entropy

OPTIMIZATIONS

↳ Fix vanishing gradient
↳ Normalize training points: center, then scale ($\mu = 0$, var = 1)
↳ Use small rand. subsample to choose ϵ
↳ Dif ϵ for dif layers of weights
↳ Emphasize schemes (repeat rare examples) (outliers ?)
↳ Acceleration scheme for SGD
↳ Add enough weights, test error ↓↓ ("double descent")



Heuristics:

① Weight decay: Add $\lambda \|w\|^2$ to loss
step $\Delta w_i = -\epsilon \frac{\partial J}{\partial w_i}$ has extra term $-2\epsilon\lambda w_i$

② Ensemble of NNs or dropout in 1 NN ($p = 0.5$)

CNN

$\text{out} = 1 + \left[\frac{\text{input} - \text{kernel height} + 2p}{\text{stride}}\right]$

① Local Connectivity → hidden unit connects to small patch from previous layer
② Shared Weights → group of hidden units share set of weights "kernel / filter"

Convolution: same linear transf. applied to different patches of input by shifting

NEAREST NEIGHBOURS

Given q, find K training points near q. dist metric up to you. Regression (avg label) or classification (majority or histogram w probabilities)

nn ↑ → underfitting | $n \to \infty$, 1-NN err $< 2B - B^2$
nn ↓ → overfitting | 2 classes, $\le 2B - B^2$

draw pts independently from some prob. distribution. $n \to \infty$, $K \to \infty$, $K/n \to 0$, err converge to B

Exhaustive K-NN Alg: Given q, scan through all n points, computing squared distances to q. Maintain a max-heap with K-shortest distances. Query time $O(nd + n \log K)$

2-5 d → Voronoi
Medium d → K-d trees
Large d → exhaustive K-NN (can use PCA / rand. proj)

Voronoi: X point set. Vor $w = \{\|p - w\| \le \|p - v\|, \forall v \in X\}$
voronoi cell always convex polyhedron / polytope
Size $\in O(n^{\lfloor d/2 \rfloor})$, often $O(n)$
For 2D: $O(n \log n)$ to compute V.d. + trapezoidal map for pt. location. $O(\log n)$ query time.

d-D: Use binary space partition tree ($\epsilon = 0 \Rightarrow$ exact, $\epsilon > 0 \Rightarrow$ approx)

K-d Trees:
1. Choose splitting feature w greatest width. ft. i in $\max(X_{ji} - X_{ki})$, make boxes cubical
2. Choose splitting value: median pt. for feature or $(X_{ji} + X_{ki})/2$. Median guarantees $O(\log n)$ depth. $O(nd \log n)$ building time. Each internal node stores a training pt.

Goal: Given q, find w so $\|q - w\| \le (1+\epsilon)\|q - s\|$
Maintain: K-nearest so far, binary min-heap of (query, stored)

UNSUPERVISED LEARNING

Sample points, no labels → discover structure in data

PCA

↳ Dimensionality reduction: find K directions that capture most of the variation
↳ Why: reducing # dimensions, remove irrelevant dimensions, find small basis for repres

X: $n \times d$, assume $\sum_i X_i = 0$, $\text{proj}_w x = \frac{x^T w}{\|w\|^2} w$

X^TX: square, symmetric, PSD, (λ_i, v_i): corresponding unit principal components

① Fit Gaussian to data with MLE

Choose K Gaussian axes of greatest variance

MLE estimates $\hat{\Sigma} = \frac{1}{n} X^TX$

1. Center X, normalize X
2. Compute unit eigenvectors of X^TX
3. Choose K, choose $v_{d-K+1} \dots v_d$
4. Compute $\vec{x} \cdot \vec{v}_i$ for each training point

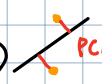
② Find direction $\vec{w}$ that maximizes sample variance of projected data

$\max_w \frac{1}{n} \sum \left(X_i \cdot \frac{w}{\|w\|}\right)^2 = \frac{1}{n} \frac{\|Xw\|^2}{\|w\|^2} = \frac{1}{n} \frac{w^T X^T X w}{w^T w}$

↳ v_d achieves max variance λ_d / n

③ Find direction $\vec{w}$ that minimizes mean squared projection distance

(proj orthogonal to proj hyperplane)



$\min_w \sum_i^n \|X_i - \tilde{X}_i\|^2 = \sum_i^n \left\| X_i - \frac{X_i \cdot w}{\|w\|^2} w \right\|^2$

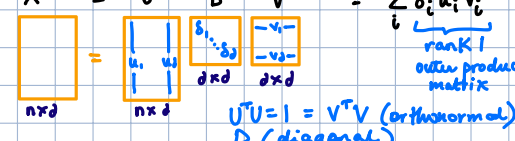
$= \sum_i^n \left(\|X_i\|^2 - \left(\frac{X_i \cdot w}{\|w\|}\right)^2 \right)$

$=$ constant $- n$ (Var from ②)

SVD

Problem: Computing X^TX takes $\Theta(nd^2)$ time, inaccurate eigenvectors

$X = UDV^T$. When $n \ge d$



$U^TU = I = V^TV$ (orthonormal)
D (diagonal)

D: $\delta_1, \dots \delta_d \to$ Nonnegative, singular values of X
Non-zero δ_i = rank (X)

$X^TX = VDU^TUDV^T = VD^2V^T$ (v_i has eigenvalue δ_i^2)

Find K greatest singular values in $O(ndK)$ time

Since $X = UDV^T \Rightarrow XV = UD \Rightarrow \text{Row}_i(UD) = X_i \cdot v_j$ (principal coordinates)

CLUSTERING

Partition data into clusters so points in clusters are more similar than across clusters

Why? Discovery, hierarchy, graph partitioning, quantization

k-means clustering: partition n points into K disjoint clusters

Cluster i mean: $\mu_i = \frac{1}{n_i} \sum X_j$

Find $\min_y \sum_i^K \sum_{y_j = i} \|X_j - \mu_i\|^2$ (squared distances from points to means)

NP-hard, Solve in $O(nK^n)$ time

K-means heuristic:

1. Fixed y_j, Update μ_i
2. Fixed μ_i, Update y_j

Halt when step (2) changes no labels. Both steps minimize cost, but not all vars. Alg never returns to prev assignment. MUST terminate. Finds local minimum

Starting:

↳ Forgy Method: Choose K rand points to be initial μ_i → Go to Step (2) better ☺

↳ Random Partition: Randomly assign pts to a cluster → Go to step (1)

Equivalent: Min within-cluster variation

$$\min_{y} \sum_{i=1}^{K} \frac{1}{n_i} \sum_{y_j=i} \sum_{y_m=i} \|X_j - X_m\|^2$$

K-Medoids Clustering:

↳ Specify distance fn $d(x,y)$

↳ Replace mean with medoid - pt that min total distance to other pts in same cluster

↳ Medoid ALWAYS input point, more robust to outliers

HIERARCHICAL CLUSTERING

K no longer a hyperparameter $O(n^3)$ time

(1) Point Set → Bottom-up 'Agglomerative' start w a point, keep fusing upwards

(2) Graphs → Top-down 'Divisive' start w everything in a cluster, keep splitting

Complete Linkage: $\max\{d(a,b)\}$

Single Linkage: $\min\{d(a,b)\}$

Average Linkage: $\frac{1}{|A||B|} \sum_{a\in A} \sum_{b\in B} d(a,b)$

Centroid Linkage: $d(\mu_A, \mu_B)$ euclidean distance

HIGH DIMENSIONS

↳ Intuition wrong in higher dim

↳ $\|p\|^2 = p_1^2 + p_2^2 + \dots p_d^2$ in d dim

$p_i \sim N(0,1)$ $p_i^2 \sim \chi^2(1)$ $E[p_i^2]=1$ $Var(p_i^2)=2$

$\|p\|^2 \sim \chi^2(d)$

↳ For large d $\|p\|$ is concentrated in thin shell around radius $\sqrt{d}$ with thickness $\propto \sqrt[4]{2d}$

↳ In high dim, 1-nn and 1000-nn don't differ much ⇒ K-means & nn not as effective

$\cos\theta = \frac{p\cdot q}{\|p\|\|q\|}$ $E[\cos\theta]=0$, $SD(\cos\theta) = \frac{1}{\sqrt{d}}$

↳ d large → cos θ tends to 0 / θ tends to 90°

↳ d large ⇒ 2 rand vectors almost always ⊥

Random Projection: Alt to PCA, preserves dist. b/w points. Project very high-d space to rand medium-d subspace

$K = \left\lceil \frac{2\ln(1/\delta)}{\varepsilon^2/2 - \varepsilon^3/3} \right\rceil$. For some point q, $\hat{q}$ is orthogonal proj of q onto S, multiplied by $\sqrt{\frac{d}{K}}$

Choosing projection direction: choose each component from a univariate Gaussian dist. then normalise vector to unit length. K rand directions, then Gram-Schmidt orthogon.

PSEUDOINVERSE

D: diagonal n×d matrix

Transpose D → replace nonzero entry w reciprocal→D^+

$DD^+D = D$ $D^+DD^+ = D^+$

SVD of $X = UDV^T$. rank D = rank X

Moore-Penrose Pseudoinv. of X: $X^+ = VD^+U^T$

1. $XX^+ = UDV^TVD^+U^T = U(DD^+)U^T$, PSD — all eigenvalues 1 or 0
2. $X^+X = VD^+U^TUDV^T = V(D^+D)V^T$, PSD
3. D, D^+, DD^+, D^+D, X, X^+, XX^+, X^+X ranks equal
4. X has rank n, $XX^+ = I_n$, X^+ right inv
5. X has rank d, $X^+X = I_d$, X^+ left inv
6. $XX^+X = X$ [$UDD^+U^TUDV^T = UDV^T = X$]
7. $X^+XX^+ = X^+$ [symmetric proof ↑]

Theorem: $X^TXw = X^Ty$; $w = X^+y$

Proof: $X^TXw = X^TXX^+y = VDU^TUDD^+U^Ty$
$= VD^2D^+V^Ty = VDV^Ty = X^Ty$

If normal eqns have multiple solns, then $w = X^+y$ is the least norm soln (minimises $\|w\|$ among all solutions)

↳ Useful when X^TX is singular

LEARNING THEORY

Range Space / Set System: Pair (P, H) where P is set of all possible test/training pts

H is set of hypotheses/classifiers

↳ $h \subseteq P$ that specifies which points h predicts are in class C

1. Power-set classifier: H power set of P containing 2^K subsets of P
2. Linear classifier: $P = \mathbb{R}^d$, H is set of all halfspaces $\{x: w\cdot x \geq -\alpha\}$ / every possible linear classifier in d dimensions

Power-set classifier sucks at learning

Risk (Generalization Error): R(h) of h is probability that h misclassifies a random point x drawn from D

∞ test data ⇒ risk = test error

Empirical Risk (Training Error): $\hat{R}(h)$ is % of X misclassified by H

$n \to \infty$, $\hat{R}(h)$ approximates R(h)

Hoeffding's Inequality:

$$Pr(|\hat{R}(h) - R(h)| > \varepsilon) \leq 2e^{-2\varepsilon^2 n}$$

Choose $\hat{h} \in H$ that minimizes $\hat{R}(\hat{h})$

Dichotomies: Dich of X is $X \cap h$, $h \in H$

For n points, 2^n dichotomies.

Dichotomy assigns points to class C or not C

If H allows all 2^n, $\hat{R}(\hat{h}) = 0$

H induces Π dichotomies.

Pr(at least one dich has $|\hat{R} - R| > \varepsilon) \leq \delta$,

$\delta = 2\Pi e^{-2\varepsilon^2 n}$

$$|\hat{R}(h) - R(h)| \leq \varepsilon = \sqrt{\frac{1}{2n}\ln\frac{2\Pi}{\delta}}$$

Smaller Π, larger n ⇒ training error ≅ true risk

smaller Π ⇒ less likely to overfit (variance↓ bias↑)

Sample complexity: # training pts needed for prob(ε) > 1−δ

$$n \geq \frac{1}{2\varepsilon^2}\ln\frac{2\Pi}{\delta}$$

SHATTER FUNCTION

of dichotomies: $\Pi_H(X) = |\{X \cap h : h \in H\}|$, $n \in |X|$

Shatter function: $\Pi_H(n) = \max_{|X|=n, X\subseteq P} \Pi_H(X)$ $\Pi_H(3)=8$ $\Pi_H(4)=14$

For all range spaces, $\Pi_H(n)$ is polynomial in n, or $\Pi_H(n) = 2^n$

Cover's Theorem: linear classifiers in $\mathbb{R}^d$ allow $\Pi_H(n) = 2\sum_{i=0}^{d}\binom{n-1}{i}$ dichotomies of n points

$n \leq d+1$, $\Pi_H(n) = 2^n$ $n \geq d+1$, $\Pi_H(n) = 2\left(\frac{e(n-1)}{d}\right)^d$

To achieve $R(\hat{h}) \leq \hat{R}(\hat{h}) + \varepsilon \leq R(h^*) + 2\varepsilon$ w $p \geq 1-\delta$:

$n \geq \frac{1}{2\varepsilon^2}\left(d\ln\frac{n-1}{d} + d + \ln\frac{4}{\delta}\right)$. Linear classifiers only need $n \in \Theta(d)$ training points for train err ≅ test err

VC DIM.

$VC(H) = \max\{n : \Pi_H(n) = 2^n\}$

H shatters set X of n pts if $\Pi_H(X) = 2^n$. VC(H) is largest X H can shatter. (X allows all 2^n dichotomies)

$\Pi_H(n) \leq \sum_{i=0}^{VC}\binom{n}{i}$. $n \geq VC(H)$, $\Pi_H(n) \leq \left(\frac{en}{VC}\right)^{VC}$

VC-dim upper bound on exponent of polynomial

$O(VC(H))$ pts. suffice for accuracy

VC finite ⇒ sample complexity grows w. # features

VC ∞ ⇒ no. amt. of training data will make it generalise well

$VC(H) = 3$. $\Pi_H(n) \leq \frac{e^3}{27}n^3$. $O(1)$ sample complexity

ADA-BOOST

ensemble, classif + regression reduces bias

Train T classifiers $G_1 \dots G_T$

correct ☺ w ↓ wrong ☹ w ↑↑

Weight for X_i in G_t grows according to how many classifiers misclassified it

Train G_t to try harder to correctly classify X_i with w↑↑

Metalearner is linear combo of learners

Test point z, $M(z) = \sum B_t G_t(z)$. G_t is ±1, M continuous. Return sign of M(z).

Iteration T: Find G_T, B_T so

$\min RiskT = \frac{1}{n}\sum_i L(M(x_i), y_i)$, $M(x_i) = \sum_{t=1}^{T} B_t G_t(x_i)$

Metalearner uses exponential loss: $L(p,l) = e^{-pl} = \begin{cases} e^{-p} & l=+1 \\ e^{p} & l=-1 \end{cases}$ emphasizes misclassifs

cost fn. for G_t = # misclassified pts

$n\cdot RiskT = \sum_i e^{-y_i M(x_i)} = \sum_i \prod_t e^{-B_t y_i G_t(x_i)}$

$= e^{-B_T}\sum w_i^{T} + e^{B_T}\sum w_i^{T}$

$w_i^{T+1} = w_i^{T} e^{-B_T y_i G_T(x_i)} = \begin{cases} w_i^T e^{-B_T} & \text{correct} \\ w_i^T e^{B_T} & \text{misclassif} \end{cases}$

B_T. Set $\frac{\partial}{\partial B_T}(RiskT) = 0$

$B_T = \frac{1}{2}\ln\left(\frac{1-err_T}{err_T}\right)$ $err_T = 0$, $B_T = \infty$ (perfect learner) $err_T = \frac{1}{2}$, $B_T = 0$

Algorithm:

1. Initialize $w_i \leftarrow 1/n$
2. For $t \leftarrow 1$ to T. i) Train G_t with weights w_i 2) Calc. err 3) Reweight points $w_i \leftarrow w_i \cdot e^{B_t}$ or $w_i \cdot e^{-B_t B_t}$,
3. Return $M\ h(z) = \text{sign}\left(\sum B_t G_t(z)\right)$

Why?: Reduces bias reliably, no hyperparameter search needed, fast, short trees + ADA = feature subset selection

linear decision boundaries don't boost well

More: $P(Y=1|x) \approx 1/(1+e^{-2M(x)})$

Exponential loss vulnerable to outliers event

If all learners beat error < 50%, train err → 0 v. fast