No.

Date

Eric Zhao's

Machine Learning 2023 Fall Note

from Prof. Shengjie Wang

Let $P(B) \neq 0$

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

Independence: $A \perp B$ iff $P(A \cap B) = P(A)P(B)$

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

Discrete: $E[X] = \sum_{x \in Val(X)} x\, p_X(x)$ (PMF)

Continuous: $E[X] = \int_a^b x f_X(x)\, dx$ (PDF)

$$Var[X] = E[(X - E[X])^2] = E[X^2] - E[X]^2$$

$$Var[a] = 0$$

$$Var[aX] = a^2 Var[X]$$

Gaussian Distribution

$$X \sim N(\mu, \sigma^2)$$

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

$$E[X^2] = \sigma^2 + \mu^2$$

No.

Date

Machine Learning 2023 Fall Note

from Prof. Shengjie Wang

Let $P(B) \neq 0$

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

Independence: $A \perp B$ iff $P(A \cap B) = P(A)P(B)$

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

Discrete: $E[X] = \sum_{x \in Val(X)} x\, p_X(x)$ (PMF)

Continuous: $E[X] = \int_a^b x f_X(x)\, dx$ (PDF)

$$Var[X] = E[(X - E[X])^2] = E[X^2] - E[X]^2$$

$$Var[a] = 0$$

$$Var[aX] = a^2 Var[X]$$

Gaussian Distribution

$$X \sim N(\mu, \sigma^2)$$

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

$$E[X^2] = \sigma^2 + \mu^2$$

No.

Date

Eric Zhao's

Machine Learning 2023 Fall Note

from Prof. Shengjie Wang

Let $P(B) \neq 0$

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

Independence: $A \perp B$ iff $P(A \cap B) = P(A)P(B)$

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

Discrete: $E[X] = \sum_{x \in Val(X)} x\, p_X(x)$ (PMF)

Continuous: $E[X] = \int_a^b x f_X(x)\, dx$ (PDF)

$$Var[X] = E[(X - E[X])^2] = E[X^2] - E[X]^2$$

$$Var[a] = 0$$

$$Var[aX] = a^2 Var[X]$$

Gaussian Distribution

$$X \sim N(\mu, \sigma^2)$$

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

$$E[X^2] = \sigma^2 + \mu^2$$

No.

Date

Likelihood:

$$L(\theta|X) = P(X|\theta)$$

e.g. $$p(x|\mu,\sigma^2) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x_i-\mu)^2}{2\sigma^2}\right)$$

Log-likelihood

$$\ell(\theta|X) = \log(L(\theta|X))$$

e.g. $$\log p(x|\mu,\sigma^2) = -\frac{1}{2\sigma^2}\sum_{i=1}^{N}(x_i-\mu)^2 - \frac{N}{2}\log(\sigma^2) - \frac{N}{2}\log(2\pi)$$

MLE: $$\mu_{ML} = \frac{1}{N}\sum_{i=1}^{N} x_i$$

$$\sigma^2_{ML} = \frac{1}{N}\sum_{i=1}^{N}(x_i-\mu_{ML})^2$$

$$E[\mu_{ML}] = \mu \qquad E[\sigma^2_{ML}] = \frac{N-1}{N}\sigma^2$$

Two Random Variables

Bivariate CDF: $F_{XY}(x,y) = P(X \le x, Y \le y)$

Bivariate PMF: $p_{XY}(x,y) = P(X=x, Y=y)$

Marginal PMF: $p_X(x) = \sum_Y p_{XY}(x,y)$

Bivariate PDF: $f_{XY}(x,y) = \frac{\partial^2 F_{XY}(x,y)}{\partial x \partial y}$

Marginal PDF $f_X(x) = \int_{-\infty}^{\infty} f_{XY}(x,y)\,dy$

Bayes':

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

No.

Date

$$E[g(X,Y)] = \iint g(x,y) f_{XY}(x,y)\,dx\,dy$$

If X, Y i.d, then $f_{XY}(x,y) = f_X(x) f_Y(y)$

$$E[XY] = E[X]E[Y]$$

$$\mathrm{cov}[X,Y] = E[(X - E(X))(Y - E(Y))] = \frac{1}{n}\sum_{i=1}^{n}(x_i - \mu_x)(y_i - \mu_y)$$

$$= E[XY] - E[X]E[Y]$$

measure how two variables change together

magnitude and direction of the linear relationship.

$$\mathrm{Var}[X+Y] = \mathrm{Var}[X] + \mathrm{Var}[Y] + 2\mathrm{cov}[X,Y]$$

$= 0$ when i.d

Pearson's correlation (normalized covariance)

$$\rho_{X,Y} = \mathrm{corr}(X,Y) = \frac{\mathrm{cov}[X,Y]}{\sigma_X \sigma_Y} \quad \in [-1,1]$$

measure the linear correlation between two variables

Multiple Random Variables.

joint distribution:

$$F_{X_1 \cdots X_n}(x_1 \cdots x_n) = P(X_1 \le x_1, X_2 \le x_2 \cdots X_n \le x_n)$$

Joint probability density function.

$$f_{X_1,\cdots X_n}(x_1,\cdots x_n) = \frac{\partial^n F_{X_1,\cdots X_n}(x_1,\cdots x_n)}{\partial x_1 \cdots \partial x_n}$$

chain rule:

$$f(x_1,\cdots,x_n) = f(x_n \mid x_1,\cdots x_{n-1}) f(x_1,\cdots,x_{n-1})$$

$$= f(x_1)\prod_{i=2}^{n} f(x_i \mid x_1,\cdots,x_{i-1})$$

If $X_1, X_2 \cdots X_n$ are independent!

$$f(x_1,\cdots x_n) = f(x_1) f(x_2) \cdots f(x_n)$$

KOKUYO

No.

Date

Covariance Matrix

$$\Sigma = \begin{pmatrix} \mathrm{Var}[x_1] & \cdots & \mathrm{Cov}[x_1, x_n] \\ \cdots & \ddots & \\ \mathrm{Cov}[x_n, x_1] & \cdots & \mathrm{Var}[x_n] \end{pmatrix} = E[(x - E[x])(x - E[x])^T]$$

Multivariant Gaussian Distribution.

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

$E[X|Y=y]$ is a function of y

$E[X|Y]$ is a random variable.

Law of total expectation:

$$E[E[X|Y]] = E[X]$$

		Mean	Var
Bernoulli (p)	$\begin{cases} p & x=1 \\ 1-p & x=0 \end{cases}$	p	$p(1-p)$
Binomial (n, p)	$\binom{n}{k} p^k (1-p)^{n-k}$	np	$np(1-p)$
Geometric (p)	$p(1-p)^{k-1}$	$\frac{1}{p}$	$\frac{1-p}{p^2}$
Poisson (λ)	$\frac{e^{-\lambda}\lambda^k}{k!}$	λ	λ
Uniform (a, b)	$\frac{1}{b-a}$	$\frac{a+b}{2}$	$\frac{(b-a)^2}{12}$
Gaussian (μ, σ^2)	$\frac{1}{\sqrt{2\pi}\sigma} e^{\frac{(x-\mu)^2}{-2\sigma^2}}$	μ	σ^2
Exponential (λ)	$\lambda e^{-\lambda x}$	$\frac{1}{\lambda}$	$\frac{1}{\lambda^2}$

No.

Date

K-means

objective: $\min_{\{s_1,\dots,s_k\}} \sum_{j=1}^{k} \sum_{x \in s_j} \|x - c_j\|^2$

without SGD: $c_j = \frac{1}{|s_j|} \sum_{x \in s_j} x$

with SGD: $c_j = (1-\eta)\, c_j + \eta x_i$ — $\sum_{j=1}^{n} (x_i - c_j)$ if $j \in \arg\min \|x_i - c_j\|^2$

$\frac{\partial f(c_1 \dots c_k)}{\partial c_j}$ 0

Initialization methods:

1) Random k-partition

2) Forgy seed method: Random start from sample points

3) K-means++ i) random start for c_1

ii) for 2.... k, choose c_j at $P = \frac{d_{j-1}(x)^2}{\sum_{i=1}^{n} d_{j-1}(x_i)^2}$

next mean is likely to be far from others

Make K-Means Faster: mini-batch | due to Parallelization

Limitation of K-Means

- No global optima guaranteed
- behave bad when clusters have varying sizes, different densities or non-spherical shapes.

No.

Date

k-Means Analysis: Connection to Gaussian Mixtures

1. Consider a uniform mixture of k spherical Gaussian distributions with means set to centroids of k-means

cov matrix is diagonal and $\Sigma = \sigma^2 I$

$$P(x) \propto \frac{1}{k} \sum_{j=1:k} \exp\left(-\|x - c_j\|^2\right)$$

Proof: $P(x \mid \mu, \sigma^2) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$

$$P(x \mid \mu, \Sigma) = \frac{1}{(2\pi)^{\frac{d}{2}} \cdot |\Sigma|^{\frac{1}{2}}} \cdot e^{-\frac{1}{2}(x-\mu)^T \Sigma^{-1} (x-\mu)}$$

for Gaussian components we assume equal variance here?

$\because$ spherical GD $\therefore \Sigma = \sigma^2 I \quad \therefore |\Sigma| = \sigma^{2d} \quad \Sigma^{-1} = \frac{1}{\sigma^2} I$

$$P(x) = \frac{1}{k} \sum_{j=1}^{k} P(x \mid \mu_j, \Sigma_j)$$

$$= \frac{1}{k} \sum_{j=1}^{k} (2\pi)^{\frac{d}{2}} \cdot (\sigma^2 I)^{\frac{1}{2}} \cdot e^{-\frac{1}{2}(x-\mu)^T (\sigma^2 I)^{-1} (x-\mu)}$$

$$= \frac{1}{k} \sum_{j=1}^{k} \frac{1}{\sqrt{2\pi} \cdot \sigma^d} e^{-\frac{1}{2\sigma^2} I \cdot \|x-\mu\|^2} \propto e^{\|x-\mu\|^2}$$

2. Let $z(x) \in \operatorname{argmin}_{c_j} \|x - c_j\|^2$ (the centroid closest to x) we have

$$\sum_{j=1}^{k} \exp\left(-\|x - c_j\|^2\right) \approx \exp\left(-\|x - z(x)\|^2\right)$$ due to exponential decay

The contribution of farther centroids is negligible

3. $\therefore P(X) \approx \text{const} \times \exp\left(-\sum_{i=1}^{n} \|x_i - z(x_i)\|^2\right)$

$\frac{dN}{dt} = -\lambda N$

Minimizing intra-cluster distances can be viewed as approximations to maximizing the likelihood

Campus Is the inductive bias here: we want the samples for k-means look like Gaussian uniform mixture?

No.

Date

K nearest neighbor classification

KNN - return the most frequent occurring class of K-nearest neighbors

$$\hat{y} = \arg\max_c \#\{i \mid y_i = c\}_{i \in NN(x,k)}$$

Learning in KNN regression is more about memorizing dataset rather than optimizing model parameters

$$MSE = \frac{1}{N} \sum_{i=1}^{N} (y^{(i)} - \hat{f}(x^{(i)}))^2$$

Important Rule:

Train V Test

Bayes Classifier:

Assume we know $P(y|x)$, we can predict label by $y^* \in \arg\max P(y|x)$

Optimal: lowest misclassification rate

No classifier can obtain a lower Bayes error

Give data X and Y, $(X|Y=c) \sim P_c$ for $c \in \{0,1\}$

Bayes error $R(h) = E_{XY}[\Pi_{h(x) \neq Y}]$

$$h^{(B)}(x) = \arg\max_{c \in \{0,1\}} P(Y=c \mid X=x)$$

excess risk: $R(h) - R(h^{(B)}) = 2\, E_X[|\eta(x) - 0.5| \Pi_{h(x) \neq h^{(B)}(x)}]$

No.

Date

KNN regressor:

$$f(x) = \frac{1}{k} \sum_{i \in NN(x,k)} y_i$$

Make KNN smooth - Kernel

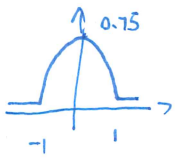
$$f(x) = \frac{\sum_{i=1}^{N} K_\lambda(x, x_i)\, y_i}{\sum_{i=1}^{N} K_\lambda(x, x_i)}$$

Epanechnikov Kernel.

$$K_\lambda(x, x_i) = D\left(\frac{|x - x_i|}{\lambda}\right)$$

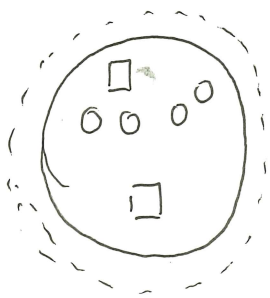
$$D(t) = \begin{cases} \frac{3}{4}(1 - t^2) & \text{if } |t| \le 1 \\ 0 & \text{otherwise} \end{cases}$$

λ determine the width of local neighbourhood

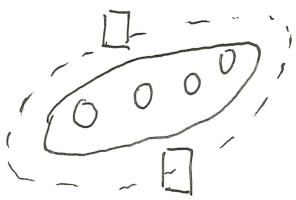


other distance metrics

Euclidean



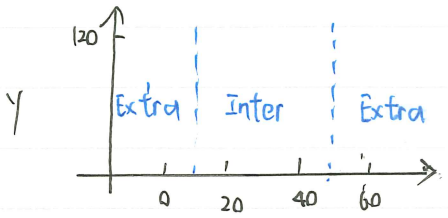
Mahalanobis Metric (马氏)



$$Dist_m(x_1, x_2, \Sigma^{-1}) = \sqrt{(x_1 - x_2)^T \Sigma^{-1} (x_1 - x_2)}$$

predict at area without samples

Extrapolation and Interpolation



3. when d↑, Euclidean distance is dominated by the biggest difference

$$\lim_{n\to\infty} E\left(\frac{dist_{max}(d) - dist_{min}(d)}{dist_{min}(d)}\right) = 0$$

4. Samples with Gaussian distribution is most likely to be found at a certain distance from the mean → concentrated on the shell

Convex Optimization

convet set C : $\forall x, y \in C$, $\theta \in [0,1]$, we have

$$\theta x + (1-\theta) y \in C$$

not convex

- R^d is convex for $\forall d$
- Affine subspace. $\{x \mid Ax = b\}$. where $A \in R^{m \times n}$, $b \in R^m$
- Polyhedron $\{x \mid Ax \leq b\}$ 多面体
- Intersection of C are C

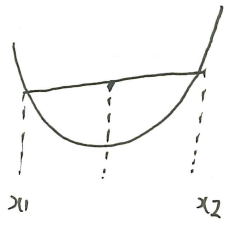
No.

Date

Convex Functions

$\forall x_1, x_2 \in D(f)$ (convex set)

$f(\lambda x_1 + (1-\lambda)x_2) \leq \lambda f(x_1) + (1-\lambda) f(x_2) \quad \lambda \in [0,1]$



- Function f is concave if $-f$ is convex
- convexity = convexity along all lines

critical point of convex function is global minimum

$f'(x) = 0$ / $f(x)$ doesn't exist

3 ways to check convexity

⓪ 0th prove by definition

$$f(\lambda x_1 + (1-\lambda)x_2) \leq \lambda f(x_1) + (1-\lambda) f(x_2)$$

① 1st: tangent line must lie below function.

$$f(x_1) + \frac{d}{dx} f(x_1)(x_2 - x_1) \leq f(x_2)$$

② 2nd: curvature is always positive

$$\frac{d^2}{dx^2} f(x) \geq 0$$

$$D_v f(p) = \lim_{h \to 0} \frac{f(p+hv) - f(p)}{h}$$

⓪ → ① directional derivative (how function changes in given direction

① → ⓪ by calculation and def of derivative

① → ② expand from $d=0$ (induction)

② → ③ by mean-value theorem.

Theorem (Stone, 1977)

Let $k \to \infty$ with increasing n s.t $\frac{k}{n} \to 0$ as $n \to \infty$.

Then for all probability distribution $P(X,Y)$, we have $R(h_{kY}^{KNN}) \to R(h^{(B)})$ optimal

The k should increase. to ensure local approximation

To find KNN

- Naive search : $O(knd)$
- tree-based data structures like Kd-tree, ball-tree (先頂区) for small d, for big d → LSH
- Approximate nearest neighbour search: locality sensitive hashing
  - similar points have higher chances to be mapped into same bucket

Pros of KNN

- simple, non-parametric
- work well in low dimensions for complex decision surfaces.

Cons of KNN

- computationally expensive, scales with data size.
- suffer from curse of dimensionality

Hyperparameters: k, distance metric, kernel

No.

Date

* $\Gamma(n) = (n-1)!$

Curse of Dimensionality

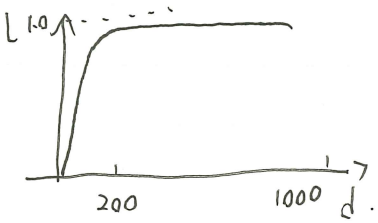
Cause:

- As d ↑, volume goes higher exponentially
- The space is much more sparse

Phenomena:

1. "Nearest neighbors" could be far away
2. More extrapolation power than interpolation
3. Euclidean distance becomes less meaningful
4. Concentration of Gaussian Distribution lies in peels rather than core

1. $l = (\frac{k}{n})^{\frac{1}{d}}$ $\quad n = \frac{k}{l^d}$



In 1000d, almost search whole space for 10 NN

2. High dimensional space is more spiky,
   All the space is at the edges
   The majority of hypersphere tend to concentrate on the shell

$$V_{ball}(d) = V_{ball}(d-1) \cdot \int_{-1}^{1} \sqrt{1-x^2}^{d-1} dx$$

$$= \frac{r^d \pi^{\frac{d}{2}}}{\Gamma(\frac{d}{2}+1)}$$

No.

Date

Checking convexity : 2nd order.

Hessian Matrix positive - semi-definite $\nabla^2_x f(x) \geq 0$

A doesn't introduce any "negative weighting"
strong enough to flip the sign. ("weighted dot product")

- (Matrix A is positive semi-definite if $x^T A x \geq 0$ for $\forall x \in R^n$)
- A symmetric matrix is p.s.d. iff all eigenvalues are non-negative

$$\nabla^2_x f(x) = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \vdots & & & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{bmatrix}$$

e.g. $H = \begin{bmatrix} 4 & 0 \\ 0 & 2 \end{bmatrix}$ 

$H = \begin{bmatrix} 4 & 0 \\ 0 & -2 \end{bmatrix}$ 

$H = \begin{bmatrix} -4 & 0 \\ 0 & -2 \end{bmatrix}$ 

$H = \begin{bmatrix} 4 & 0 \\ 0 & 0 \end{bmatrix}$ 

the kernel of svm must be psd
covariance matrix is psd

for this.
check whether Hessian is
p.d / p.s.d for x
to avoid saddle

No.

Date

Convex Functions:

Exponential: $f(x) = e^{ax}$ for $\forall a \in R$

Affine function $f(x) = a^T x + b$ where $x \in R^n$.

both convex and concave; line in 2D and plane in 3D.

Quadratic Functions: $f(x) = \frac{1}{2} x^T A x + b^T x + c$.

for a symmetric matrix $A \in R^{n \cdot n}$ and $b \in R^n$ and $c \in R$.

Nonnegative weighted sums of convex functions $f(x) = \sum_{i=1}^{k} \underbrace{w_i} f_i(x)$ ≥ 0

Jean's inequality:

Extend from convex function, we have

$$f\left(\sum_{i=1}^{k} \lambda_i x_i\right) \leq \sum_{i=1}^{k} \lambda_i f(x_i) \quad \text{for } \sum_{i=1}^{k} \lambda_i = 1 \text{ and } \lambda_i > 0$$

extend to integrals:

$$f\left(\int p(x) x \, dx\right) \leq \int p(x) f(x) \, dx \quad \text{for } \int p(x) \, dx = 1 \text{ and } p(x) \geq 0$$

$$f(E[x]) \leq E[f(x)]$$

Gradient Descent

$$x_{t+1} = x_t - \eta \nabla f(x_t)$$

Linear Regression

$$y = w_0 + w_1 x + \varepsilon$$

$$MSE = L(w) = \frac{1}{n} \sum_{i=1}^{n} e_i^2 = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{w}_0 - \hat{w}_1 x_i)^2$$

We want $\frac{\partial}{\partial w} L(w) = 0$.

$$\therefore \hat{w}_1 = \frac{\sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n} (x_i - \bar{x})^2}, \quad \hat{w}_0 = \bar{y} - \hat{w}_1 \bar{x}$$

No.

Date

Multivariable:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - w_0 - w_1 x_{i,1} \dots w_p x_{i,d})^2$$

$$= \frac{1}{n}(y - Xw)^T(y - Xw) \quad \text{Convex}$$

Goal (differentiate MSE): $-X^T 2(y - Xw) = 0$

Multicollinearity: one predictor variable in a multiple regression can be linearly predicted from the others.

- Sensitive to small changes in data.
- redundant variable
- with multicollinearity, X is not full-rank.
  X^TX can't be inverted.
  OLS $(X^TX^{-1})X^Ty$ doesn't exist
  least square solution not unique.

SGD: More iterations, but cheaper

Lower overall cost. Mini-batch SGD achieve better trade-off

Minimizing MSE = Maximizing likelihood

$l(w) = \log L(w|x)$

$$= \sum_{i=1}^{N}\left[-\log(\sqrt{2\pi}\sigma) - \frac{(y_i - w^Tx_i)^2}{2\sigma^2}\right.$$

$$= -N\log(\sqrt{2\pi}\sigma) - \frac{1}{2\sigma^2}\sum_{i=1}^{N}(y_i - w^Tx_i)^2$$

KOKUYO

No.

Date

Summary of supervised learning

- Assume data is generated i.i.d from unknown distribution P

$$(x_i, y_i) \sim P(X, Y)$$

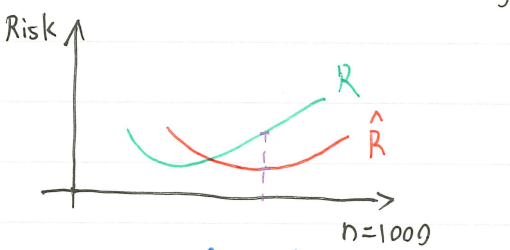
- The goal is to minimize expected error (true risk) under P

$$R(w) = \int P(x,y)(y - w^T x)^2 dx\,dy = E_{x,Y}[(y - w^T x)^2]$$

- Estimate the true risk by empirical risk

$$\hat{R}_D(w) = \frac{1}{|D|} \sum_{(x,y) \in D} (y - w^T x)^2$$

We don't want to optimize on training data.



Generalization error

we want to solve $w^* = \arg\min_w R(w)$

not $\hat{w}_D = \arg\min_w \hat{R}_D(w)$

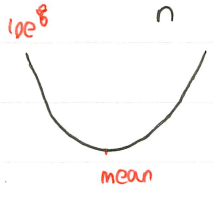
Potential Problem: Outliers, High Leverage Points

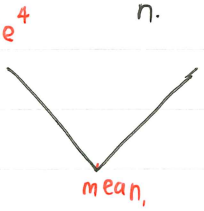
No.

Date

$$MSE = \frac{\sum_{i=1}^{n} (y_i - y_i^{(p)})^2}{n}$$

$$MAE = \frac{\sum_{i=1}^{n} |y_i - y_i^{(p)}|}{n}$$



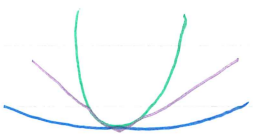


punish less for outliers

Huber Loss (Smooth Mean Absolute Error)

$$L_\delta(y, f(x)) = \begin{cases} \frac{1}{2}(y - f(x))^2 & \text{for } |y - f(x)| \le \delta \\ \delta|y - f(x)| - \frac{1}{2}\delta^2 & \end{cases}$$

MSE when residual is small



$\delta = 10$

$\delta = 1$

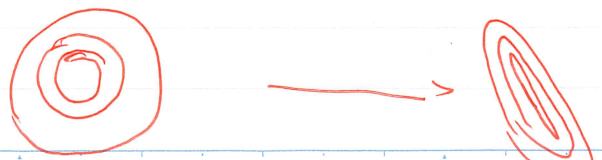
$\delta = 0.1$

Potential Problem - Collinearity

- Consider the level set $\{w | (w - \hat{w})^T X^T X (w - \hat{w}) = C\}$

The equation of an ellipsoid (X full rank), the lengths of the axes scales with eigenvalues $(X^T X)^{-1}$

- When two predictors are highly correlated, the contours of MSE run along a narrow valley, broad range for coefficient estimate
- With x_1 and x_2 linearly related, $X^T X$ has a 0 eigenvalue
- So the level set $\{w | (w - \hat{w})^T X^T X (w - \hat{w}) = C\}$ is no longer an ellipsoid. it's a degenerated ellipsoid., where contour lines are pair lines



hard to get proper coefficients

No.

Date

Decide important variables.

: identify a subset of suitable variables

Forward stepwise selection.

: begins with a model containing no predictors
then add predictors that gives greatest additional improvement

Backward stepwise selection,

- begins with full least squares model.
iteratively removes the least useful predictor

Potential problem. : synergy between two predictors

$Y: w_0 + w_1 x_1 (1 + w_2) x_2$

$Y: w_0 + w_1 x_1 + w_2 x_2 + w_3 x_1 x_2$.

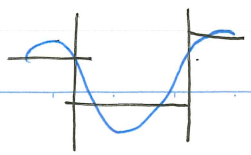
interaction term

Polynomial Regression:

$$\begin{aligned} Y &= w_0 + w_1 x + w_2 x^2 + \cdots + w_D x^D \\ &= w_0 + w_1 \phi_1(x) + w_2 \phi_2(x) + \cdots + w_D \phi_D(x) \\ &= w^T \phi(x) \end{aligned}$$

Piecewise polynomial :

Represent the function by a separate polynomial in each interval



The function could be discontinuous

Campus

X^TX: structure of data

$(X^TX)^{-1}$: structure of model

No.

Date

Linear Regression:

Compared to the population, our Least Square Estimation

$$\hat{w} = (X^TX)^{-1}X^Ty$$

is a random variable, even though ground truth w^* is not

Bias and Variance Decomposition

suppose the true model is $f^*(x) = E[Y|x]$

Our estimate: expected prediction over all possible datasets

$$f_D(x) = \text{argmin}_f \frac{1}{n}\sum_{i=1:n}(y_i - f(x_i))^2.$$

$$\text{error}(f) = E_{p(x)}[(f(x) - E[Y|x])^2] - E_{p(x,y)}[(E[Y|X] - y)^2]$$

error in prediction; noise, irreducible error, nonrelated

$$E_D[(f_D(x) - E[Y|x])^2] = E_D[(f_D(x) - \bar{f}(x))^2] + E_D[(\bar{f}(x) - E[Y|x])^2]$$

variance; bias

$$E_x(\text{squared loss}) = E_x\{[E_D(f(x) - \hat{f}(x))]^2 + \text{Var}_D[\hat{f}(x)] + \text{Var}_\varepsilon(\varepsilon)\}$$

Assume data is generated using a linear model: $y = Xw^* + \varepsilon$.

ε is random and i.i.d. with mean 0 and variance σ^2 $E_D(w) = w^*$

The bias is 0: $E_D[x^T\hat{w}] = x^Tw^*$ or $E_D[\hat{w}] = w^*$

The variance is $E_D[(x^T\hat{w} - x^Tw^*)^2] = \sigma^2 x^T(X^TX)^{-1}x$

The variance scales with σ^2, and roughly with cov matrix

$\leftarrow$ d and $\frac{1}{n}$ $\quad$ n ↑ Var↓ Poverfit)↑

$\text{trace}(xx^T)$ $\quad$ d ↑ Var↑ Poverfit)

roughly eigenvalue of $(X^TX)^{-1}$

Model complexity ↑ bias ↓ variance ↑

OLS has smallest error because smallest variance

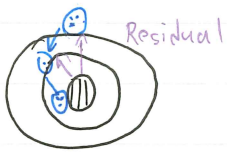
1+3+5

No. 7-3

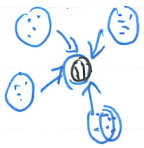
Date

* bootstraping : 有放回的多次抽样

Bagging : deal with high variance
Boosting: deal with high bias



in-series model
Boosting



parallel model
Bagging

Ridge Regression:

In Regression Setting

$$\min \sum_{i=1}^{n} (y^{(i)} - \sum_{j=1}^{d} x_j^{(i)} w_j - w_0)^2 \quad \text{s.t} \sum_{j=1}^{d} w_j^2 \le R.$$

$\equiv$ Lagrange Multiplier

$$\min \sum_{i=1}^{N} (y^{(i)} - \sum_{j=1}^{p} x_j^{(i)} w_j - w_0)^2 + \lambda \sum_{j=1}^{N} w_j^2$$

- one to one correspondence between λ and R.
- Intercept w_0 is left out of penalty
- Penalization of L2 norm is weight decay

拉格朗日算子:

函数 $f(x,y)$ 与约束 $g(x,y)$ 相切时有极值，切点法向量平行且梯度方向相同但梯度大小可能不同，故引入乘子 $\lambda \ge 0$

$$\nabla f(x,y) = \lambda \nabla g(x,y)$$

函数的梯度 = 约束函数的梯度的向量方程

No.

Date

Logistic Regression: $\sigma(z) = \frac{1}{1+e^{-z}} = \frac{e^z}{e^z+1}$

$$p(x) = \frac{e^{w^T x}}{1+e^{w^T x}}$$

$$\frac{p(x)}{1-p(x)} = e^{w^T x} \quad \text{odds}$$

$$\log\left(\frac{p(x)}{1-p(x)}\right) = w^T x \quad \text{Logit}$$

Maximum Likelihood Estimation.

In MLE, we want to maximize the conditional probability of observing the data given a specific probability distribution and parameters

For binary classific: $P(Y=1|x) = \frac{1}{1-e^{-w^T x}}$

$$P(Y=0|x) = 1 - \frac{1}{1-e^{-w^T x}}$$

$$\therefore P(Y=y|X=x) = p(x)^y (1-p(x))^{1-y}$$

max log-likelihood: $l(w) = \sum_{i=1}^{n} [y_i \log p(x_i; w) + (1-y_i) \log (1-p(x_i; w))]$

= min Cross-Entropy Loss: $L(w) - \sum_{i=1}^{n} [y_i \log p(x_i; w) + (1-y_i) \log (1-p(x_i; w))]$

Entropy: For a discrete event Y: $H(Y=y_0) = -\log(p(y_0))$

For the random variable Y: $H(Y) = -\sum_{i=1}^{k} p(y_i) \log p(y_i)$

Intuition of quantifying information:

Measuring how much surprise there is in an event

Those events that are rare have more information than those common events

No.

Date

Cross Entropy

$$H(p,q) = -\sum_{i=1}^{k} p(y_i)\log q(y_i)$$

Expected surprisal of an observer with subjective probabilities Q upon seeing data that were actually generated according to probabilities P

$H(p,q)$ is minimized when $P = Q$

KL 散度

$$H(p,q) = H(p) + D_{KL}(p\|q)$$

$$D_{KL}(p\|q) = -\sum_{y} p(y)\log\frac{q(y)}{p(y)}$$

$$\frac{\partial L(w)}{\partial w_j} = -\sum_i [y_i - \underbrace{p(x_i;w)}_{\sigma(z)}]\, x_{ij}$$

Check convexity: $\sigma'(z) = \sigma(z)(1-\sigma(z))$

$$\nabla^2 L(w) = \sum_i x_i x_i^T p(x_i;w)(1-p(x_i;w))$$

$x_i x_i^T$ is P.S.D. since $(x_i x_i^T)x_i = (x_i^T x_i)x_i$ and $(x_i^T x_i) \geq 0$

Newton-Raphson algorithm: to find θ s.t. $f(\theta) = 0$

$$\theta \leftarrow \theta - \frac{f(\theta)}{f'(\theta)}$$

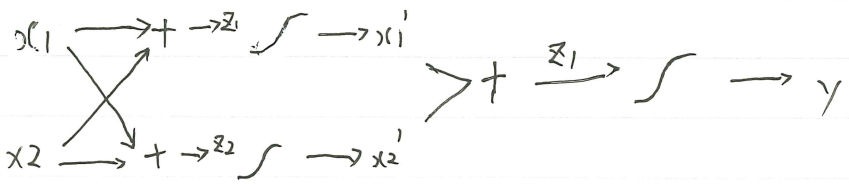
∴ For optimization at $f'(\theta) = 0$

$$\theta \leftarrow \theta - \frac{f'(\theta)}{f''(\theta)}$$

By taylor expansion: $0 = f(a) = f(\theta^t) + f'(\theta^t)(a-\theta^t) + \frac{1}{2}f''(\theta^t)(a-\theta^t)^2$

$$\frac{f(\theta^t)}{f'(\theta^t)} + (a-\theta^t) = \frac{-f''(\theta^t)}{2f'(\theta^t)}(a-\theta^t)^2 \longrightarrow a-\theta^{t+1} = \frac{-f''(\theta^t)}{2f'(\theta^t)}(a-\theta^t)^2$$

Logistic Regression is well calibrated.



$$E_D[(f_D(x) - E[Y|x])^2] = E_D[A^2] + E_D[B^2]$$
$$\simeq E_D[(f_D(x) - \bar{f}(x))^2] + E_D[(\bar{f}(x) - E[Y|x])^2]$$

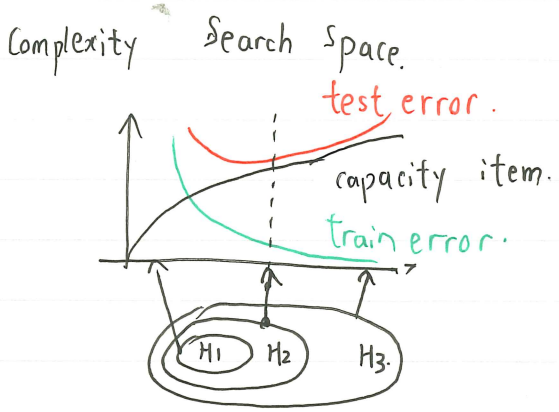
Variance — Bias.

Difference : model trained from random data — mean model

mean model — data distribution.

Gauss-Markov Theorem: Among all unbiased estimators, OLS estimator has the smallest variance and hence has the smallest error of all

$$Var(x^T \hat{w}) \leq Var(h_{linear\_unbiased}(x))$$



No.

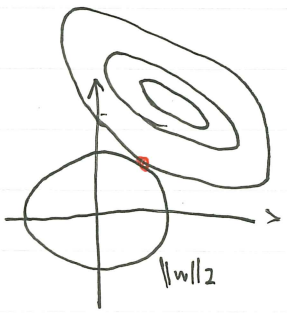
Date

Ridge Regression Lagrange Multiplier:

$\therefore$ Optimal is achieved when

$$\nabla(L(w) + \lambda \|w\|_2^2) = 0$$

- The point where two countours are tangent to each other



- standard least squares coefficient estimate is scale-equivalent

multiply X by c $\rightarrow$ scaling w^T by $\frac{1}{c}$

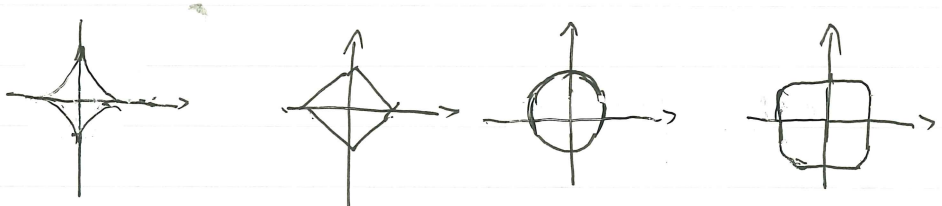
- Ridge regression is NOT scale-equivalent

we usually standardize before apply ridge

close-form solution: $-\frac{1}{n}\sum_{i=1}^{n} x_i \left(1_{[y_i=y]} - \frac{e^{z_i}}{e^z}\right) + \eta W$

Lasso Regression:

$$\min \sum_{j=1}^{n} \left(y_i - \sum_{j=1}^{d} x_{i,j} w_j - w_0\right)^2 + \lambda \sum_{j=1}^{d} |w_j|$$



$$\|w\|_p = \left(\sum_{j=1}^{d} |w_j|^p\right)^{\frac{1}{p}}$$

No.

Date

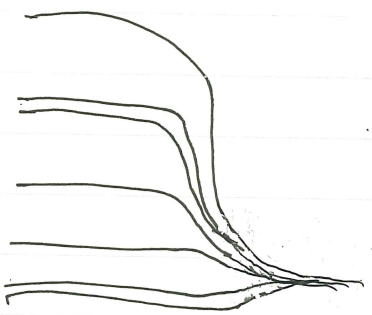
$$\frac{\partial \|w\|_1}{\partial w_j} = \begin{cases} 1 & \text{if } w_j > 0 \\ -1 & \text{if } w_j < 0 \\ ? & \text{if } w_j = 0 \end{cases}$$

subgradient: For a convex function f. a vector g is a subgradient of f at x. if for all z we have $f(z) \geq f(x) + g^T(z-x)$

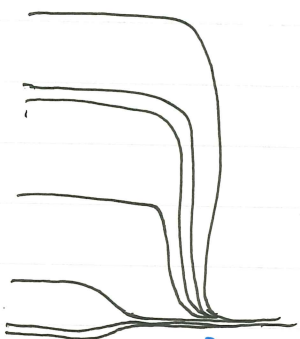
- A convex function (not necessarily differentiable) is minimized at x. if the zero-vector is a subgradient at x.

$$0 \in \partial f(x^*) \text{ iff } x^* \in \operatorname{argmin}_x f(x)$$

Ridge $\|w\|_2$



Lasso $\|w\|$



$a_j = \sum_i x_{ij}^2$, $c_j = 2\sum r_{ij} x_{ij}$

$$w_j = \begin{cases} \frac{c_j + \lambda}{a_j} & \text{if } c_j < -\lambda \\ 0 & \text{if } |c_j| \leq \lambda \\ \frac{c_j - \lambda}{a_j} & \text{if } c_j > \lambda \end{cases}$$

soft thresholding

$$\text{sign}(c_j) \max(0, |c_j| - \lambda) / c_j$$

No.

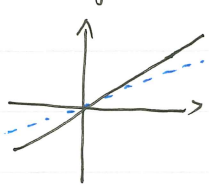
Date

Estimator	Formula
Feature Section	$\hat{w}_j \cdot I(\hat{w}_j > \hat{w}_{(M)})$
Ridge	$\hat{w}_j / (1+\lambda)$
Lasso	$\text{sign}(\hat{w}_j)(\lvert\hat{w}_j\rvert - \lambda)_+$

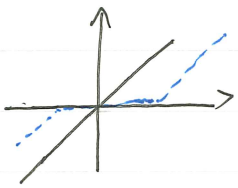
proportional shrinkage

translates coefficient by λ, truncating at 0

Ridge

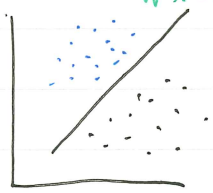


Lasso



SVM (Support Vector Machine)

find the largest margin between classes of points



$$G(x) = \begin{cases} -1 & w^T x + w_0 < 0 \\ +1 & w^T x + w_0 > 0 \end{cases}$$
$$= \text{sign}(w^T x + w_0)$$

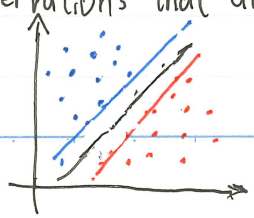
Hyperplane: $w^T x + x_0 = 0$ with normal vector w

Margin: Smallest distance from samples to hyperplane

Maximal Margin classifier — maximal margin hyperplane

Support Vectors: 3 training observations that are equidistant from the maximal margin hyperplane

The rest samples are irrelevant



No.

Date

for Hyperplane $w_0 + w^T x = 0$ · normal vector is $\bar{w} = \frac{w}{\|w\|_2}$

Signed distance of any point x to h:

$$\bar{w}^T (x - x_0) = \frac{1}{\|w\|_2} (w^T x + w_0)$$

$$= \frac{1}{\|w\|_2} f(x)$$

Optimization Objective:

$$\max_{w, w_0} M \qquad \min_{x_i \in X} \text{dist}(x_i, h) \geq M$$

$$\text{s.t. } \forall i \quad \frac{y_i}{\|w\|_2} (w^T x_i + w_0) \geq M.$$

for one optimal solution w^* and w_0^*

Cw^* and Cw_0^* is also an optimal solution.

A Trick: set M to $\frac{1}{\|w\|_2}$

$\therefore \min \|w\|_2$ (maximize $\frac{1}{\|w\|_2}$)

Another trick: $\min_{w, w_0} \frac{1}{2}\|w\|_2^2$

$$\text{s.t. } \forall i \cdot y_i (w^T x_i + w_0) \geq 1$$

For Lagrange Multiplier, a more abstract form:

$\min r(w)$

$$\text{s.t. } \forall i \; g(w) \leq 0$$

Lagrangian: $L(w, a) = r(w) + \sum_{i=1}^{n} a_i g_i(w)$

penalty weights

Lagrangian multipliers

g_i and r are convex, and solutions satisfy Karush-Kuhn-Tucker conditions

KOKUYO

No.

Date

Primal $\min_{w} \max_{a_i \geq 0} L(w, a)$

Inner Optimization.

Given a fixed value w. we want to vary α. s.t. maximize the penalty for violating the constraint

Outer Optimization.

After inner optimization. , we vary w to minimize the loss given the best efforts of α controller

Dual: $\max_{a_i \geq 0} \min_{w} L(w, a)$

Karush- Kuhn - Tucker Conditions (KKT)

$$L(w, a) = r(w) + \sum_{i=1}^{n} a_i g_i(w)$$

$$\begin{cases} \frac{\partial L}{\partial w}(w, a) = 0 \\ a_i g_i(w) = 0 \\ g_i(w) \leq 0 \\ a_i \geq 0 \end{cases}$$

saddle point is optimal

Two scenarios

① at solution $g_i(w^*) < 0$, it's optimized in an unrestricted way, so $\frac{\partial r}{\partial w}|_{w^*} = 0$, $a_i^* = 0$

② at $g_i(w^*) = 0$, $\exists a_i^*$ s.t. $\frac{\partial L}{\partial w}(w^*; a_i^*) = \frac{\partial r}{\partial w}|_{w^*} - a_i^* \frac{\partial g_i}{\partial w}|_{w^*} = 0$. $a_i^* > 0$

No.

Date

Dual Problem of SVM:

Primal - duel problem

- different perspective
- Easier to solve

$$= \frac{1}{2}\|w\|^2 - \sum_{i=1}^{n} a_i [y_i (w^T x_i + w_0) - 1]$$

Dual problem: $\max \min L(w, w_0, a)$

Set $\nabla_w L(w, w_0, a) = \nabla_{w_0} L(w, w_0, a) = 0$, we get

$$\begin{cases} w = \sum_{i=1}^{n+m} a_i y_i x_i & ① \\ 0 = \sum_{i=1}^{n} a_i y_i & ② \end{cases}$$

$$\|w\|_2^2 = \langle w, w \rangle = \langle \sum_i a_i y_i x_i, \sum_i a_i y_i x_i \rangle$$

$$= \sum_i a_i y_i \sum_j a_j y_j \langle x_i, x_j \rangle$$

$$= \sum_i \sum_j a_i a_j y_i y_j \langle x_i, x_j \rangle$$

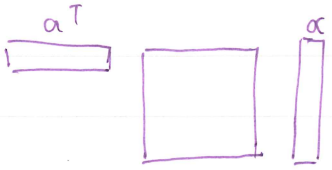
plug in ①, ② $\therefore L(w, w_0, \alpha) =$

$$\sum_i a_i - \frac{1}{2} \sum_i \sum_j a_i a_j y_i y_j \langle x_i, x_j \rangle$$

= A form of Quadratic Programming

$$\min \frac{1}{2} a^T Q a + c^T a$$

$$\text{s.t. } Aa \le b$$



$Q_{ij} = y_i y_j \langle x_i, x_j \rangle$

- more general objective is convex if Q P.S.D
- hard to get global minimum for general case.
- use sequential minimal optimization to solve problem globally

SMO: slove for two coordinates at a time (similar to coordinate descent!)

Support Vectors:

From KKT, the constraints $a_i[y_i(w^T x_i + w_0) - 1] = 0$

① $a_i > 0 \quad y_i(w^T x_i + w_0) = 1$ x_i is on the boundary

② $y_i(w^T x_i + w_0) > 1 \quad a_i = 0$ x_i is not on boundary

$$\therefore f(x) = w^T x + x_0$$

$$= w_0 t \sum_{x' \in [S.V]} y_i a_i \langle x', x \rangle$$

Line of thought

1. Change constrained optimization into Lagrangian form.

$$L(w, a) = \underbrace{r(w)}_{\text{obj.}} + \underbrace{\sum_{i=1}^{n} a_i g_i(w)}_{\text{constraints}}$$

2. convex $\longrightarrow$ primal and dual problem are equivalent
3. study dual problem $\max_{a_i \geq 0} \min_{w, w_0} L(w, w_0, a)$
4. KKT conditions $\longrightarrow$ simplified to a quadratic programming problem

$$\max \sum_{i=1}^{n} a_i - \frac{1}{2} \sum_{i,j} y_i y_j a_i a_j \langle x_i, x_j \rangle$$ (to predict based on S.V)

5. slove it by quadratic programming solvers

No.

Date

Soft Margin SVM

if two classes are not separable

sample can be on the wrong side of margin or hyperplane

- greater robustness to individual observations
- better classification for most observations

$$\max_{w, w_0, \xi_1 \cdots \xi_n} M$$

subject to $\frac{y_i}{\|w\|_2}(w_0 + w^T x_i) \geq M(1-\xi_i)$ for $i = 1, \cdots n$.

slack variables $\xi_i \geq 0$ allow observations to lie on the wrong side

$\sum_{i=1}^{n} \xi_i \leq \lambda$

! hyperparameter

$$\min_{w, w_0} \frac{1}{2}\|w\|_2^2 + C\sum_i \xi_i$$ when $C \uparrow$, more S.V.

subject to $y_i(w_0 + w^T x_i) \geq (1-\xi_i)$ for $i = 1, \cdots n$

$\xi_i \geq 0$

$\max(1 - y_i(w_0 + w^T x_i), 0) = \xi_i$

If not violating the margin: $y_i(w_0 + w^T x_i) \geq 1$.

- same as the hard margin SVM case. the constand 1 here comes from the trick that we force the norm of w to be $\frac{1}{M}$
- Therefore $\xi_i = 0$

If violating the margin, ξ_i indicates how much it violates

If with GD $\nabla_w L(x_i, y_i) = C[\mathbb{I}(1 - y_i(w^T x_i + w_0) > 0)(y_i x_i)] - w$

$\nabla_{w_0} L(x_i, y_i) = C\mathbb{I}(1 - y_i(w^T x_i + w_0) > 0)\, y_i$

$$\mathbb{I}(Q) = \begin{cases} 1 & Q \text{ is True.} \\ 0 \end{cases}$$

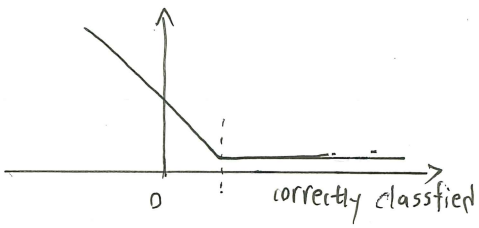
KOKUYO

No.

Date

Hinge Loss $\min_{w, w_0} \frac{1}{2}\|w\|_2^2 + C\sum_i \max(1 - y_i(w_0 + w^T x_i), 0)$

$\hookrightarrow L(x_i, y_i, w, w_0) = \begin{cases} 0 & \text{if } 1 - y_i(w_0 + w^T x_i) \le 0 \\ 1 - y_i(w_0 + w^T x_i) & \text{otherwise} \end{cases}$



soft Margin SVM uses L2 regularized hinge loss

Kernel trick: $\phi(x) = (x_1, x_1^2 \dots x_1^D, x_2, x_2^2 \dots, x_2^D, x_1x_2, x_1x_2^2, x_1x_2^{D-1} \dots)$

Give $\phi_D(\cdot)$ for polynomial of exactly D-degree.

$\phi_D(u) \cdot \phi_D(v) = k_D(u, v) = (u \cdot v)^D$

∴ in SVM $f(x) = w_0 + \sum_{(x', y') \in \{S.V.\}} y' \langle x', x \rangle$

$= \langle w, \phi(x) \rangle + w_0$

$= w_0 + \sum_{(x', y') \in \{S.V.\}} y' k(x', x)$

The dual formulation of SVM allow us to us kernal trick → capability↑

No.

Date

Linear kernel: $k(x_i, x_j) = x_i^T x_j$

d-degree polynomial $k(x_i, x_j) = (1 + x_i^T x_j)^d$

Radial: $k(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2)$

↑ tiny $k(x_i, x_i)$ when test observation x is far from training observation x_i

Radial basis function kernel is like doing a KNN with smooth neighbourhood

let $\gamma = \frac{1}{2}$ $\exp(-\frac{1}{2}\|x_i - x_j\|^2)$

$$= \exp\left(-\frac{1}{2}(\|x_i\|_2^2 + \|x_j\|_2^2)\right) \exp(\langle x_i, x_j \rangle)$$

$$= C \sum_{n=0}^{\infty} \frac{\langle x_i, x_j \rangle^n}{n!}$$

No.

Date

Decision Trees

Tree structure. $\mathcal{T} \triangleq (N \cup L, E)$

L: set of leaves N: set of nodes other than other leaves

E: edges of the tree (directed)

$$\theta = \{\mathcal{T}, \{w_j\}_{j \in L}, \{(p_j, \tau_j)\}_{j \in N}\}$$

$\{w_j\}_{j \in L}$: model used at every leaf

$\{(p_j, \tau_j)\}_{j \in N}$ splitting function at every node in N

feature dimension to split, splitting threshold

piece-wise constant: constant value in every region.

piece-wise linear: compute $w_j^T x$ in every region

final decision function of decision tree:

$$f(x) = w_{q(x)}$$

$q(x)$: tree assignment function: it traverse the tree with input x and return the leaf node

when adding 2 more nodes, the objective change from $\min \sum_i (y_i - w_0)^2$ (τ = Mean(Y))

to $\min_{p_0, \tau_0} \sum_{(x,y) | x_{p_0} \le \tau_0} (y - w_1)^2 + \sum_{(x,y) | x_{p_0} > \tau_0} (y - w_2)^2$

How to solve? : For every feature dimension p, we sort all data points based on that dimension, and try every middle value as the split, return the best one

No.

Date

When to stop? 1. Max depth / min samples per leaf (hyperparameters)

2. when gain = $L_{n-1} - L_n$ is to small

Optimize a tree for classification

prediction can be a probability vector given the assigned data points

$$w_j = \frac{1}{|R_j|} \sum_{(x,y) \in R_j} \text{onehot}(y, C)$$

a C-length one-hot vector with position y = 1

Pred class $\leftarrow$ $\arg\max_{i=0:C-1} w_{j,i}$

If w_j is one-hot (all training data points of j has same label), then the entropy is minimized

If w_j is uniform. $w_{j,i} = \frac{1}{C}$ for all i. the entropy is maximized

Objective for splitting:

$$\max_{p,\tau} H(w) - \left(\frac{|R_{left}|}{|R|} H(w_{left}) + \frac{|R_{right}|}{|R|} H(w_{right}) \right)$$

Information Gain: $IG(X) = H(Y) - H(Y|X)$

i.e. with more knowledge about the conditioning features, how much entropy gets reduced

How to solve? Search. find the one with largest decrease

No.

Date

Learning the simplest decision tree is an NP-complete problem

Normally we use greedy approach

1. split current node with largest gain., do this recursively
2. No guarantee on resulting tree.
3. Fast and simple

Good things:

- Easily interpretable
- mirror human decision-making.
- Display graphically
- Trees are piecewise constant / linear models.
  very capable in fitting the data. (might overfit)

Other tricks: pruning: remove node s.t. decrease in performance is minimal

Random Forest.

- Generate K single trees with randomness, average prediction as final

$$\frac{1}{K}\sum_k f_k(x)$$

Bootstrap: randomly sampling from population to construct new training set

Aggregating (Bagging) Use average output $\frac{1}{K}\sum_k f_k(x)$ as final aggregated predict

No.

Date

Why RF performs better than bagged trees?

- It incorporates a small tweak that decorrelates individual trees
  This further reduces variance when we average trees
- The tree may only split on a predictor
  from a randomly selected subset of m predictors (features)
- A fresh selection of m randomly selected predictors is presented for each split of each tree

$m = \sqrt{d}$ turns out to be a good choice.

e.g. we have 100 predictors, each split we choose 10

RF and bagging —— ensemble method.

Gradient Boosting Decision Trees

- newly trained tree corrects the error of the previous trees

$$\hat{y}_i^k = \hat{y}_i^{k-1} + f_k(x_i)$$

parameters of tree k

penalize number of leaves and norm of w^k

regularization $= \gamma|L_k| + \frac{1}{2}\lambda\|w^k\|_2^2$

$$F_k(\theta_k) = \sum_{i=1}^{n} l(y_i, \hat{y}_i^k) + \Omega(\theta_k) = \sum_{i=1}^{n} l(y_i, \hat{y}_i^{k-1} + f_k(x_i)) + \Omega(\theta_k)$$

overall loss

local quadratic approximation

with taylor's expansion: $l(y_i, \hat{y}_i^{k-1} + f_k(x_i)) \approx l(y_i, \hat{y}_i^{k-1}) + g_i f_k(x_i) + \frac{1}{2} h_i f_k^2(x_i)$

with k-1 tree fixed, constant in optimization.

$$\therefore F_k(\theta_k) + \text{const.} \approx \sum_{i=1}^{n} \left[g_i f_k(x_i) + \frac{1}{2} h_i \cdot f_k^2(x_i) \right] + \gamma|L_k| + \frac{1}{2}\lambda\|w^k\|_2^2$$

$$= \sum_{i=1}^{n} \left[g_i w^k_{q_k(x_i)} + \frac{1}{2} h_i \cdot (w^k_{q_k(x_i)})^2 \right] + \gamma|L_k| + \frac{1}{2}\lambda\|w^k\|_2^2$$

$w^k_{q_k(x_i)}$: output / weight of leaf node $q_k(x_i)$

w^k is a vector containing all leaf weights.

w^k_j : leaf weight for leaf j

KOKUYO

No.

Date

I_j: all the data points assigned to I_j

we change from summing over data point to summing over leaves

$$F_k(\theta_k) + \text{const} \simeq \sum_{j \in L_k} \left[\left(\sum_{i \in I_j} g_i \right) w_j^k + \frac{1}{2} \left(\lambda + \sum_{i \in I_j} h_i \right) (w_j^k)^2 \right] + \gamma |L_k|,$$

$$\therefore \frac{\partial F_k(\theta_k)}{\partial w_j^k} = \left(\sum_{i \in I_j} g_i \right) + \left(\lambda + \sum_{i \in I_j} h_i \right) w_j^k$$

$\therefore$ for optimal w_j^k $\quad w_j^k = \frac{-G_j}{H_j + \lambda}$, $G_j \triangleq \sum_{i \in I_j} g_i$, $H_j \triangleq \sum_{i \in I_j} h_i$

Now we only need to know I_j, so we need to search over to set ρ_j, τ_j

Loss after split on node j:

$$F_k(\theta_k; w_j^k(L), w_j^k(R), \{w_{j'}^k\}_{j' \in L_k \setminus j}) = \gamma(|L_k| + 1) - \frac{1}{2} \sum_{j \in L_k \setminus j} \frac{G_{j'}^2}{H_{j'} + \lambda} - \frac{1}{2} \frac{G_{j(L)}^2}{H_{j(L)} + \lambda}$$

$$-\frac{1}{2} \frac{G_{j(R)}}{H_{j(R)} + \lambda}$$

$$\therefore IG = \frac{1}{2} \left[\frac{G_{j(L)}^2}{H_{j(L)} + \lambda} + \frac{G_{j(R)}^2}{H_{j(R)} + \lambda} - \frac{(G_{j(L)} + G_{j(R)})^2}{H_{j(L)} + H_{j(R)} + \lambda} \right] - \gamma$$

$\therefore$ GBDT Algorithm Overview

1. Pick $\hat{y}^0$ e.g. average of all labels
2. From $k = 1$ to K
   - ① For (x_i, y_i), compute $g_i \triangleq \frac{\partial \ell(y_i, \hat{y}_i^{k-1})}{\partial \hat{y}_i^{k-1}}$, $h_i \triangleq \frac{\partial^2 \ell(y_i, \hat{y}_i^{k-1})}{(\partial \hat{y}_i^{k-1})^2} = \frac{\partial g_i}{\partial \hat{y}_i^{k-1}}$ (current position)
   - ② Build tree k by recursively optimizing improvement by splitting node j

$$\max_{(\rho_j, \tau_j)} \frac{1}{2} \left[\frac{G_{j(L)}^2}{H_{j(L)} + \lambda} + \frac{G_{j(R)}^2}{H_{j(R)} + \lambda} - \frac{(G_{j(L)} + G_{j(R)})^2}{H_{j(L)} + H_{j(R)} + \lambda} \right] - \gamma$$

Campus

No.

Date

Boosting: Combine multiple weak classifier (generated sequentially)
use a different loss function or
reweight training samples (to solve previous mistakes)

Square error: $L(y, f(x)) = (y - f(x))^2 = (e_{k-1,i} - a_k g(x_i; \gamma))^2$
learn residual of the current model, more robust to outliers

Exponential loss: $L(y, f(x)) = \exp(-y f(x))$
$u_i^{(k)} = \exp[-y_i f_{k-1}(x_i)]$ = apply weight to each data point
make model keep learning after training error is 0

Adaboost:
Idea: Train $g_2(x)$ on a new training set that fails $g_1(x)$

ε_1: error rate of $g_1(x)$

$$\varepsilon_1 = \frac{\sum_{i=1}^{N} u_i^1 I(y_i \neq g_1(x_i))}{\sum_{i=1}^{N} u_i^1} \qquad \varepsilon_1 < 0.5.$$

u_i^k: weight for sample i in training classifier k

Change the example weights from u_i^1 to u_i^2 s.t.

$$\frac{\sum_{i=1}^{n} u_i^2 I(y_i \neq g_1(x_i))}{\sum_{i=1}^{N} u_i^2} = 0.5.$$

as confusing as possible

Train g_2 based on new weights u_i^2

No.

Date

e.g. (x_1, y_1, z_1) $u_1^{(1)} = 1$ ✓ $u_1^{(2)} = \frac{1}{\sqrt{3}}$

(x_2, y_2, z_2) $u_2^{(2)} = 1$ ✗ $u_2^{(2)} = \sqrt{3}$

(x_3, y_3, z_3) $u_3^{(3)} = 1$ ✓ $u_3^{(2)} = \frac{1}{\sqrt{3}}$

(x_4, y_4, z_4) $u_4^{(4)} = 1$ ✓ $u_4^{(2)} = \frac{1}{\sqrt{3}}$

$\varepsilon_1 = 0.25$ $\varepsilon = 0.5$

$g_1(x)$ $g_2(x)$

$g_1(x) = y$: $u_i^{(2)} \leftarrow u_i^{(1)} / d_1$

$g_1(x) \neq y$: $u_i^{(2)} \leftarrow u_i^{(1)} * d_1$ more weight for misclassified classes

let $\sum_{g_1(x_i)=y_i} \frac{u_i^{(1)}}{d_1} = \sum_{g_1(x_i)\neq y_i} u_i^{(1)} \cdot d_1$ $d_k = \sqrt{(1-\varepsilon_k)/\varepsilon_k}$

1. Initialize observation weights $u_i^{(1)} = \frac{1}{n}$
2. From $k = 1$ to K.
   - Fit classifier g_k to data.
   - compute ε_k
   - update weights
     ① misclassified $u_i^{(k+1)} \leftarrow u_i^{(k)} d_k$
     ② Else: $u_i^{(k)} \leftarrow u_i^{(k-1)} / d_k$
3. Output $f(x) = \text{sign}\left(\sum_{k=1}^{K} \alpha_k g_k(x)\right)$

No.

Date

$w_i^{(k)}$

∵ for every iteration : $w_i^{(k+1)} = w_i^{(k)} \cdot \exp(-y_i g_k(x_i) a_k)$

$$\therefore w_i^{(K)} = w_i^{(1)} \prod_{k=1}^{K-1} \exp(-y_i g_k(x_i) a_k)$$

$$= \exp\left(-y_i \sum_{k=1}^{K-1} g_k(x_i) a_k\right)$$

$$= \exp(-y_i f_{k-1}(x_i))$$

a_k

$$a_k = \arg\min_{\alpha} \sum_i e^{-y_i [f_{k-1}(x_i) + \alpha g_k(x_i)]}$$

$L(f_{k-1} + \alpha g_k)$

$$\frac{\partial L}{\partial \alpha} = \sum_i y_i g_k(x_i) e^{-y_i [f_{k-1}(x_i) + \alpha g_k(x_i)]} = 0$$

$$-\sum_{i: g_k(x_i) = y} e^{-y_i f_{k-1}(x_i) - \alpha y_i g_k(x_i)} + \sum_{i: g_k(x_i) \neq y} e^{-y_i f_{k-1}(x_i) - \alpha y_i g_k(x_i)} = 0$$

$$-\sum_{i: g_k(x_i) = y} w_i^{(k)} e^{-\alpha} + \sum_{i: g_k(x_i) \neq y} w_i^{(k)} e^{\alpha} = 0$$

$$-(1-\varepsilon_k) e^{-\alpha} + \varepsilon_k e^{-\alpha} = 0$$

$$a_k = \frac{1}{2} \ln \frac{1-\varepsilon_k}{\varepsilon_k} = \ln\left(\sqrt{\frac{1-\varepsilon_k}{\varepsilon_k}}\right) = \ln(d_k)$$

adaboost factor : $\eta_k = \log \sqrt{(1-\varepsilon_k)/\varepsilon_k}$

No.

Date

Naive Bayes (Basically just doing counts)

$$\arg\max_Y P(Y=y) \prod_j P(x_j = x_j \mid Y = y)$$

(If no x_j, add some fake counts)

Point metric:

	Label positive	Label negative
Predict Positive	TP	FP
Predict negative	FN	TN

$$\text{Recall} = \frac{TP}{TP+FN} \qquad \text{Precision} = \frac{TP}{TP+FP}$$

F1-score:

$$F_1 = \frac{2 \cdot \text{Recall} \cdot \text{Precision}}{\text{Recall} + \text{Precision}}$$ (harmonic mean)