PRML

Evaluation:

Assignments - 30%

Mini Project - 30%

Mid Term - 20%

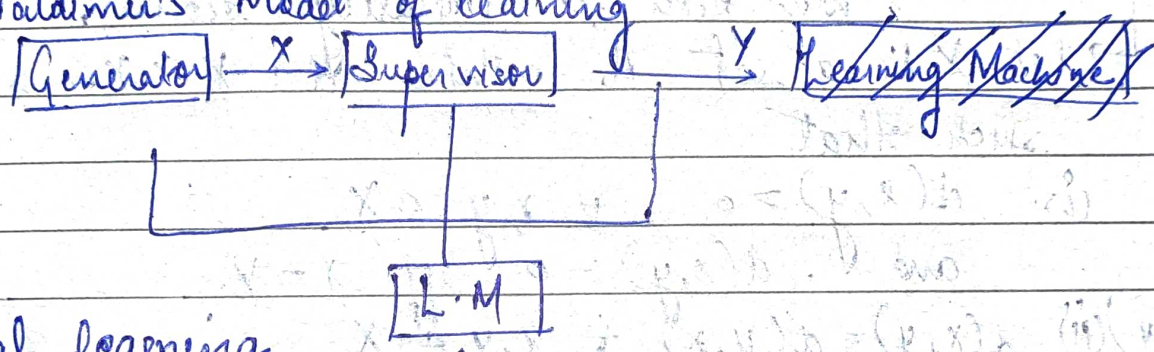
End Term - 20%

Presentation -

[LabWork - In Matlab]
↳ Every wednesday //

Intro:

Vaidimir's Model of learning



Empirical learning Model

20/01/22

↳ learning machine Zaki - PCA
Vapnik - SVM
Friedman - Bias
Weisenwith - Mathematics

Classification Problem:

$T = \{(x_1, y_1), \ldots, (x_l, y_l)\}$

$x_i \in \mathbb{R}^n$, $y_i \in \{1, 2, \ldots, K\}$
↳ K classes.

Knowing 'f', that defines relationship b/w X and Y.

We believe mathematics is all about determinism. It is not stochastic.

Number System
— Natural, Whole, Rational, ... Real.
This will be used. ↑

Operations on Real Numbers
$+, *$.
$(R, +, *)$ → Called field in algebra.

Metric Space
eg Model audio signals.
↳ We need similarity scores.
↳ Let X be a set. A metric $d: X \times X \to R$ such that

(Positivity) (i) $d(x,y) > 0$, $\forall x, y \in X$.
and $d(x,y) = 0$ iff $x = y$.

(Symmetricity) (ii) $d(x,y) = d(y,x)$ $\forall x, y \in X$.

(iii) Triangle Inequality.
$d(x,y) + d(y,z) \geq d(x,z)$ $\forall x, y, z \in X$.

$A = \{(x_1, \ldots, x_n) : x_1, x_2 \ldots x_n \in \mathbb{R}\}$
↳ Set of in ordered tuples.

Manhattan Distance
$$d(x,y) = \sum_{i=1}^{n} |x_i - y_i|$$

Euclidean Distance
$$d(x,y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} = \sqrt{(x-y)^T (x-y)}$$

Infinity Norm Distance
$$d(x,y) = \max_{i=1,\ldots n} (|x_i - y_i|)$$ (Induced by infinity norm)

Majorly used in statistics

Mahalonobis Distance → (X is how far from Y in terms of std deviation)

$$d(X,Y) = X^T \Sigma^{-1} Y$$

↳ Σ^{-1} is positive semidefinite matrix

$$= [x_1 \ \ldots \ x_n] \begin{bmatrix} \sigma_{11} & \sigma_{12} & \ldots & \sigma_{1n} \\ \vdots & & & \\ \sigma_{n1} & \ldots & & \sigma_{nn} \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$$ (Σ^{-1})

= (+ve) value because Σ is the semidefinite matrix.

$$= \sum_{i=1}^{n} \left(\frac{x_i - y_i}{\sigma_i}\right)^2 = \sum_{i=1}^{n} \frac{(x_i - y_i)^2}{\sigma_i^2}$$

Addition: Audio Signal X + Audio Signal Y } Intro to Vector Space.

Multiplication by scalar: $\alpha(\quad) \to$

PAGE No
DATE: / 202

42:00

Def: Vector Space and prop. of V.S:

$F = (\mathbb{R}, +, *)$
↑ field

1. Commutative
2. Associative
3. Additive identity.
4. Multiplicative identity.
5. additive inverse.
6. Scalar multiplication is associative.
7. Scalar multiplication is distributive

Eg $\mathbb{R}^n$ is a V.S.

Another Eg. V.S - Collection of all $n \times n$ matrix with real values.

Def: Subspace:

Def: Linear independence

Prop. → A finite set of vectors is linearly dependent iff one of them can be expressed as linear combinations of them others.

- Basis of V.S - Set of vectors which are linearly independent & spans V is called a basis of V.

- Inner Product Space:
  ↳ 2 vectors inputs, then it gives a scalar value.

norm of a / length of (a) = $\langle a, a \rangle$.

TA: AnandPD

PAGE No
DATE: / 202

25/01/22

L-3

$V = \{ R \to R \}$ | ECG signal
$f+g$, αf | ↳ functional data analysis.
↳ classify whether this has disease or not.

→ Def: Span.
→ Basis:
→ Equivalent: • B is basis of V.
• B is minimal gen. set.

→ Std. Basis for $\mathbb{R}^3$, $\mathbb{R}^n$ → to
Linear Mapping → linearity property satisfied.

→ (Coordinates) →
→ Coordinate vector / coordinate representation.

$$\begin{bmatrix} 1 & 2 & 6 \\ 5 & 1 & 3 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 5 \end{bmatrix}$$

→ linear transformation.

→ Norm
→ Euclidean Norm (aka l_2 norm).
→ Manhattan (aka) (l_1 → norm).

→ Infinite norm.
→ General inner product.
↳ Bineal mapping omega.
→ dot product = Std inner product.
↳

** If we are not getting proper results with std inner product (i.e dot product) then we can switch to different inner product. This gives rise to kernel methodes.
for getting linearly separable.

- Eigen values & eigen vectors
- Any inner product induce the norm.
  $\|x\|_2 = \sqrt{x^T x}$.
  ↳ further are certain norms not induced by any other inner product.
  ↳ Eg $\|x\|_1$.

distance b/w x & y
$d(x,y) := \|x-y\|_2 = \sqrt{\langle x-y, x-y \rangle}$

Gradient & derivatives
Gradient vector → vector of partial derivatives w.r.t of every component.

Convex Sets.
$C \subseteq \mathbb{R}^n$ is convex, if

Convex functions.

Unconstrained convex optimization.

Properties.

If f_1 & f_2 are convex, then
$\max(f_1, f_2)$ is also convex
$f_1 + f_2$ are also convex functions.

If a function is convex, then, it becomes very easy to find gradients.

Gradient is still a vector.

Distribution & density functions.

Regression Problems

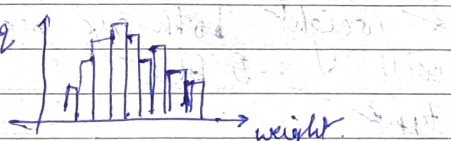
mean → most commonly occuring value.

$y_i = E[Y/X_i] + \epsilon_i$
↳ dist^n of y_i & mean.
↓
target function → we want to find a function, that approximate this.

$E[\epsilon_i] = 0$

Assumption behind regression model.

27/03/22



Group the data & then plot the histogram.

We can also Plot relative freq.

Actual freq / Total points × BinSize

for plotting we divide by binsize as well.

Area of histogram should ~~be equal~~ represent relative frequency of that class.

R.f. * BinSize.
= Af / T. Points

Not a good method
↳ soln, reduce the size of the bin.

total no

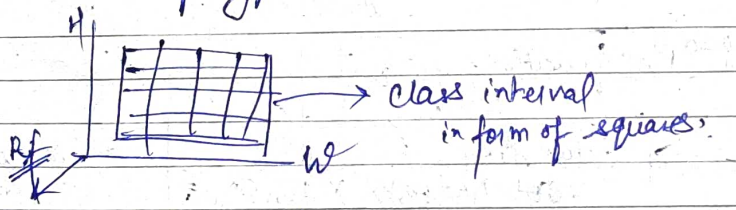
Relative freq of people between a weight 60 to 100,
↳ Area under the curve b/w 60 to 100.

↳ Better to represent using some distributions
→ Area under curve.

$$\int_{60}^{100} f(x)\,dx.$$

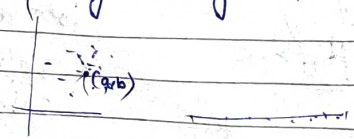
Law of large numbers: $n \to \infty$, the relative freq. converts to probability.

When height & weight - both are present
↳ we will 3-D plot
↳ landscape type



- Multivariate normal distribution.
- Unconstrained convex optimization.

$f: R^2 \to R$
$f(x,y) = x^2 + y^2$ # move in dire. where changee in functional value should be maximum



Rate of change in direction of some unit vect $\vec{u}$
↳ $\vec{u} \cdot \nabla f(a,b)$ (dot product of $\vec{u}$ & gradient of f at (a,b))

$$\boxed{\arg\max_{\vec{u}} \nabla f(x)^T u}$$

find a unit vector y such that
$x^T y$ is max.
then $y = \frac{x}{\|x\|}$ — $x^T x$ □

$$u = \frac{\nabla f(x)}{\|\nabla f(x)\|}$$

↳ If we move in direction of gradient, then rate of change of f will be max

$$\frac{\nabla f(x)^T \nabla f(x)}{\|\nabla f(x)\|} = \frac{\|\nabla f(x)\|^2}{\|\nabla f(x)\|}$$

$$\Rightarrow \|\nabla f(x)\|$$

Ascent → ↳ magnitude of change

dir of Descent

$$u = -\frac{\nabla f(x)}{\|\nabla f(x)\|}$$

Gradient descent algorithm.

↳ If our function is convex, then we are guaranteed to converge to global minima, using gradient descent method.

Theorem

$\forall$ any $L \geq 0$;

$$\|\nabla f(x) - \nabla f(y)\|_2 \leq \|x - y\|_2$$

then ~~we should~~ this equality

$$f(x^k) - f(x^*) \leq \frac{\|x^{(0)} - x^{(*)}\|_2^2}{2tk}$$

$t = \gamma$ // step length

$x^k \rightarrow$ kth iteration. $\quad x^* \rightarrow$ optimal solution

$x^0 \rightarrow$ initial.

Stochastic G.D.

Regression Model

↳ finding functional dependency of X & Y

$Y_i = f(x_i) + \varepsilon_i$

$\downarrow$

$E[Y/X]$

↳ This can be obtained when we have knowledge of all data i's given, but in general we don't have all data points

Mean regression model

(II) Quantized Reg. Models

$$P(Y \leq f(x)) = \tau \quad (\text{tau})$$

~~Empirical Risk minimi~~

Mean Regression Problem:

$$Y = f^*(x) + \varepsilon_i$$

$\downarrow$

$E[Y|X]$

Types loss function $\rightarrow$ 1) $L(f(x_i) - Y) = (f(x) - Y)^2$

2) $L(f(x) - Y)) = |f(x) - Y|$

diff types of errors

↳ Before modelling we decide the loss function.

$$\Sigma L(f(x_i) - Y_i) = \int L(f(x_i) - Y_i)\, dP(x, y)$$

relative freq.

↳ Sum of all losses should be minimum.

$dP(Y/x)\, P(x)$

Paradigm - 1.

Paradigm - 2 Parametric distr. model.

$$= \frac{1}{L} \sum_{i=1}^{L} L(Y_i - f(x_i))$$

Limited Data

// Only for L train points.

$$= \frac{1}{L} \Sigma (Y_i - f(x_i))^2$$

But for $L \to \infty$ (By laws of large numbers)
the relative freq. converge to
actual probability.

ERM principle

$$\int L(Y_i - f(x_i))\, dP(x,y)$$

$$= \int (Y_i - f(x_i))^2\, dP(x,y).$$

SRM
↳ Structural Risk Minimization.

Original prob.

$x \xrightarrow{f} y$

$L \to$ loss function.

$f \in F$ = {Linear funct - - -}

F_2 = { quadratic - - - }

F = { polynomial }

$x \in R^n$, $y \in R$

$w_1x_1 + w_2x_2 + \dots + w_nx_n + b$ → linear function in R_n

$= w^T x + b$

$$w = \begin{bmatrix} w_1 \\ w_2 \\ w_n \end{bmatrix}$$

$$x = \begin{bmatrix} x_1 \\ x_n \end{bmatrix}$$

$F = \{ w^T x + b : w \in R^n, b \in R \}$

$L = (Y - f(x))^2 = u^2$

where $u = Y - f(x)$

↳ least square loss function

↳ adv → if it convex & also

$$\min_{w,b} \frac{1}{L} \sum (y_i - (w^T x_i + b))^2$$

$w \in R^n$, $b \in R$

least square linear reg. model.

Date
01-02-2022

L-4

Normal dist. density function:

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

% of data	Interval
68% →	$[\mu-\sigma, \mu+\sigma]$
95% →	$[\mu-2\sigma, \mu+2\sigma]$
99.7% →	$[\mu-3\sigma, \mu+3\sigma]$

(estimate of mean) $\bar{x} = \frac{1}{n} \sum_{i=1}^{n} (x_i, y_i)$

Sample Variance $= \frac{1}{n-1} \sum_{i=1}^{n} (x_i - \bar{x})^2$

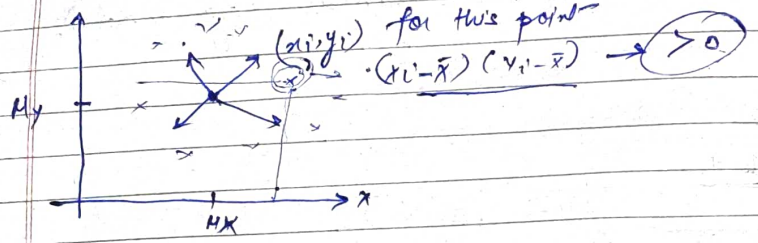
} 1 D.

In 2-D, we have to calc, covariance.

$\Sigma \to$ Covariance matrix.

$$\Sigma = \begin{bmatrix} \sigma x^2 & \sigma_{xy} \\ \sigma_{xy} & \sigma y^2 \end{bmatrix}$$

$$Cov(x,y) = \frac{1}{n-1}\sum_{i=1}^{n}(x_i-\bar{x})\cdot(y_i-\bar{x})$$



Variance in particular dir d :→ where d is unit vector.

$$\hat{d}^T \Sigma \hat{d}$$

Want to calc. gradient $\nabla \cdot d$

H & W both following Normal dist.

Project all points on any axis.
↳ And it follows normal dist
↳ Then it will also follow normal dist
↳ We can say x & y jointly follow normal dist.

Multivariate normal dist.

3-D Case

$$X = (x_1, x_2, \ldots, x_n) \sim N(\mu, \Sigma)$$
$$w^T X \sim N(w^T \mu, w^T \Sigma w)$$

$x = (x, y)$

$x = (x_1, x_2)$

$f(x) = f(x_1)\, f(x_2)$.

vector

$$= \frac{1}{\sqrt{2\pi}\sigma} \exp -\frac{(x-\mu_1)}{\sigma_1^2} \cdot \frac{1}{\sqrt{2\pi}\sigma_2} \exp -\frac{(x-\mu_2)}{\sigma_2^2}$$

$$= \frac{1}{(\sqrt{2\pi})^2 |\Sigma|^{1/2}} \exp -\frac{1}{2}(x-\mu)^T \Sigma^{-1} (x-\mu).$$

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \qquad \Sigma = \begin{bmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_2^2 \end{bmatrix} \qquad \mu = \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}$$

$Cov(x,y) = 0$. (if x & y are independent).

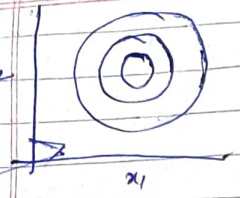
If x & y are dependent,

Density function of normal dist (Gaussian dist) in N-dim.

$$\frac{1}{\sqrt{2\pi}^n |\Sigma|^{1/2}} \exp -\frac{1}{2}(x-\mu)^T \Sigma^{-1} (x-\mu).$$

$x = (x_1, \ldots, x_n)$

Countours :- points where $f(x) = c$

① 

$f(x_1, x_2)$
$f(x_1, x_2) = c$

If $var(x) = var(y)$ → then there will be co-centric. ~~circle~~ ellipse

If x & y are independent, then - circle should be there.
↳ $Cov.(x,y) = 0$
and $Var(x) = Var(y)$.

② Multivariate distr.
Gradient descent
Convex Optimization.

08/02/2022
09/02/2022

Least Square Regression Model

$T = \{(x_1, y_1), \ldots, (x_2, y_2)\}$
$x_i \in \mathbb{R}^n$ and $y_i \in \mathbb{R}$

$x \xrightarrow{f} y$

Regression Model
$y_i = f(x_i) + \epsilon_i$
$E[\epsilon_i] = 0$

$\sum_{i=1}^{L} L(y_i - f(x_i))$ → should be minimum.

PAGE No
DATE : / / 202

↳ least square loss function.

$L(x, y, f) = (y_i - f(x_i))^2$ → Convex

We need to minimize →

Q1

$\min_{f} \sum_{i=1}^{L} (y_i - f(x_i))^2$
↳ Empirical loss ↳ as we have selected only known points

$\int L(x, y, \hat{x}) \, dP(x, y)$
↳ Minimize this when we have all data points.

Q2 → Where to select this "f".

Linear function in n dimension can be obtained by
$f \in (w \cdot$

∴ Minimization problem reduces to.

$\min_{w,b} \sum_{i=1}^{L} (y_i - (w^T x_i + b))^2$ → Convex prob. & smooth func of w & b

$w_1 x_1 + w_2 x_2 + \ldots + w_n x_n + b = 0$ ← hyperplane

→ $w_1 x_1 + w_2 x_2 + b = 0$.
→ $w_1 x_1 + w_2 x_2 + w_3 x_3 + b = 0$.

$\nabla f(x) = 0$.

$\begin{bmatrix}1\\2\end{bmatrix}[1 \quad 2]$, $2\times20 \quad 20\times2$ (2×2)

$Y = \begin{bmatrix} y_1 \\ y_2 \\ y_L \end{bmatrix}$ → L data points.

$A = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1n} & 1 \\ x_{21} & \cdots & \cdots & x_{2n} & 1 \\ x_{L1} & & & x_{Ln} & 1 \end{bmatrix}$ → added last col. of 1's ; L → datapoints.

$u = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_{n-1} \\ w_n \\ b \end{bmatrix}$

$(Y - Au)$

$= (Y-Au)^T(Y-Au)$

→ Equivalent to

$\text{Min} \sum_{i=1} \left(Y_i - (w^T x_i + b) \right)^2$

$Y_i - A_i u = Y_i - w_1 x_{i1} + w_2 x_{i2} + \cdots + w_n x_{in} + b$

Problem Reduces to.

$\min_u \; \underbrace{(Y-Au)^T(Y-Au)}_{h(u)}$

where $u = \begin{bmatrix} w \\ b \end{bmatrix}$ $w \in \mathbb{R}^n$, $b \in \mathbb{R}$.

$\nabla_u h(u) = 0.$

$\left[(-A)^T (Y-Au) + (Y-Au)^T(-A) \right] = 0$

$= -A^T Y + A^T A u + -Y^T A + u^T A^T A = 0$

=

M=1

(A) = [ ] 20x2

$(Y-Au)^T (Y-Au)$

$= (Y^T - u^T A^T)(Y - Au)$

$= Y^T Y - Y^T A u - u^T A^T Y + u^T A^T A u$

These are same.

$= Y^T Y - 2u^T A^T y + u^T A^T A u.$

$2A^T(Y - Au) = 0.$

$= A^T Y - A^T A u = 0$

$= A^T A u = A^T y$

$\Rightarrow u = (A^T A)^{-1} A^T y$

→ Least square solution.

$A^T(Y - Au) = 0$

(A) = $\begin{bmatrix} 1 & x_1 & x_1^2 \\ 1 & & \\ 1 & & \\ 1 & & \end{bmatrix}$ M=2

$\begin{bmatrix} x_1^2 & x & 1 \\ x_2^2 & x_2 & 1 \end{bmatrix}$ 20x3 ; train 50x3

eg

$A = \begin{bmatrix} 3 & 3 & 1 \\ 5 & 6 & 1 \\ 2 & 4 & 1 \\ 4 & 9 & 1 \end{bmatrix}$ $y = \begin{bmatrix} 78 \\ 112 \\ 56 \\ 176 \end{bmatrix}$

$u = \begin{bmatrix} w_1 \\ w_2 \\ b \end{bmatrix} = (A^T A)^{-1} A^T Y = \begin{bmatrix} 6.30 \\ 16.91 \\ -9.59 \end{bmatrix}$

b → Biased termed.

Quadratic function in $\mathbb{R}^n$.

$f(x) = w_{11} x_1^2 + w_{21} x_2^2 + \cdots + w_{n1} x_n^2$

$+ w_{12} x_1 x_2 + w_{13} x_1 x_3 + \cdots + w_{1n} x_1 x_n$ ← many cross terms

$+ c_1 x_1 + c_2 x_2 + \cdots + c_n x_n + b$

Quadratic terms ↑ linear terms →

$$f(x) = x^T H x + c^T x + b$$

$$H = \begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1n} \\ w_{21} & w_{22} & & w_{2n} \\ \vdots & & & \\ w_{n1} & \cdots & & w_{nn} \end{bmatrix}$$

$$C = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{bmatrix}$$

Least square quadratic regression model.

$\text{Min}_{(w)} f(w) = \text{Min}_{(w)}$ 4-

Want to obtain polynomial function of degree m and $x_i \in \mathbb{R}^n$

x_1^m
x_2^m
⋮
x_n^m
$x_1 x_2^{m-1}$
⋮
$x_n x_n^{m-1}$
⋮
$x_1^2 x_2^{m-2}$

x_1^3
x_2^3
⋮
x_n^3
$x_1 x_2^2$
⋮
x_1
x_2
x_5
x_n

Matrix $\phi \rightarrow \begin{bmatrix} \phi_1(x) \\ \phi_2(x) \\ \vdots \\ \phi_{n-1}(x) \end{bmatrix}$

$$w = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_{n-1} \\ w_0 \end{bmatrix}$$

$\text{Min}_{(w)} (Y - w^T \phi)^T (Y - w^T \phi)$

$w = (\phi^T \phi)^{-1} \phi^T Y$. $\quad \phi = \begin{pmatrix} \phi_1(x_1) & \cdots & \phi_{M_\phi}(x_1) \\ \vdots & & \\ \phi_1(x_N) & & \phi_{M_\phi}(x_N) \end{pmatrix}$

$\phi_1(x) = 1$
$\phi_2(x) = 2$ — $\phi_1(x) = x^3 x_2^2$

M- degree polynomial fit in N- dimension.

$x_i \in \mathbb{R}^1$

Assignment $\quad Y_i = \sin(2\pi x_i) + \underbrace{\epsilon_1}_{\text{Noise}}$

$$RMSE = \sqrt{\frac{1}{L} \sum_{i=1}^{50} (\hat{Y}_i - Y_i)^2}$$

↳ calc

Bishop → 2nd Chapter

10/02

Linear Estimate

$f(x) = w^T x + b$

find $n+1$ parameters — $w \in \mathbb{R}^n$, $b \in \mathbb{R}$.

$w = (x^T x)^{-1} x^T y$

Non-linear Estimate

(M1) $f(x) = w^T \phi(x) + b$

$\phi(x)$ → matrix consisting of $M-1$ basis function.

$$\phi(x) = \begin{bmatrix} \phi_1(x) \\ \vdots \\ \phi_{M-1}(x) \end{bmatrix}$$

(M2) $f(x) = w^T \phi(x)$.

where $\phi(x) = \begin{bmatrix} \phi_1(x) \\ \vdots \\ \phi_M(x) \end{bmatrix} = I$

$w = (\phi^T \phi)^{-1} \phi^T y$

What happens when we move from linearity to non-linearity.

x → $\phi(x)$ → function of x.

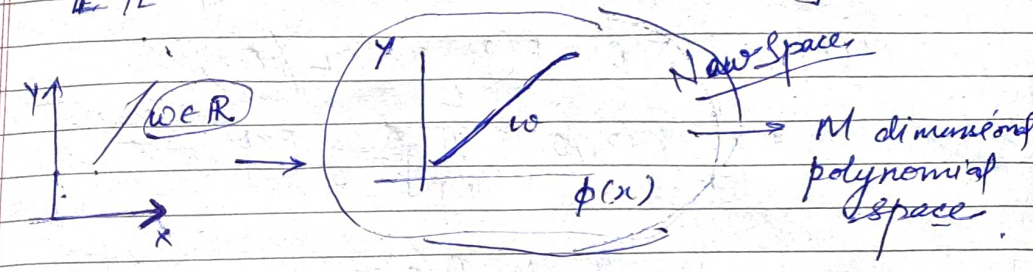
All x's are mapped to a new space where also linearly separable.

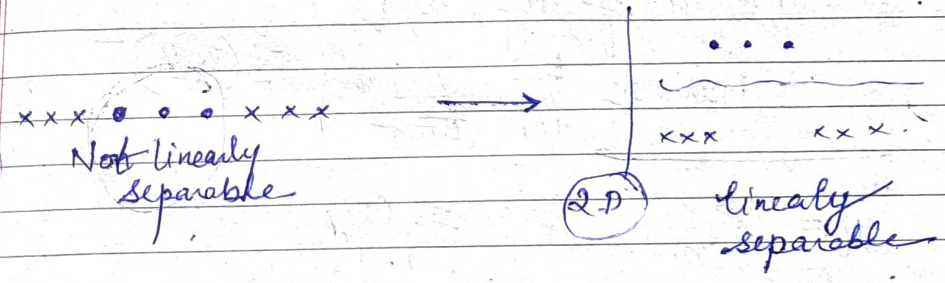
$\phi_i(x)$ → are function.

Infinite dimensional Hilbert space.

and from here, we will study about kernels.

$$\phi = \begin{bmatrix} \phi_1(x_1) & \phi_2(x_1) & \cdots & \phi_M(x_1) \\ \phi_1(x_2) & & & \\ \vdots & & & \\ \phi_1(x_L) & \cdots & \cdots & \phi_M(x_L) \end{bmatrix}$$





Eg

$$\phi(x) = \begin{bmatrix} 1 \\ x \\ x^2 \end{bmatrix}$$

$f(x) = w_2 x^2 + w_1 x + w_0$.

Gaussian Basis function.

$\phi_j(x) = e \cdot \exp\left(-\frac{1}{2s_j^2}(\quad)\right)$

We can obtain any curve by combining basis with appropriate parameters.

Distribution (D) → (θ) → D(θ)

$\{x_1, x_2 \_\_\_\_\_ x_L\}$

Q: what is T here?

$P(\theta/T) = \frac{P(T/\theta)\,P(\theta)}{P(T)}$ (likelihood), Priori - probability

$P(\theta/T) = P(T/\theta) = P(x_1, x_2, x_L/\theta)$

$= \prod_{i=1}^{L} P(x_i/\theta) = \prod_{i=1}^{L} \frac{1}{\sqrt{2\pi}\sigma} \exp^{-1/2\left(\frac{x-\mu}{\sigma}\right)^2}$

Max, take derivative w.r.t θ

Maximization of $\log P(\theta/T)$

$= \text{Max} \log \prod_{i=1}^{L} \frac{1}{\sqrt{2\pi}\sigma} \exp^{\frac{-1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$

$\mu = \frac{\sum x_i}{n}$ $\quad \sigma = \frac{\sum (x_i - \mu)}{n-1}$

Maximum likelihood estimate of θ w.r.t T

Q Is → $E[Y/X]$ → same as $f(x)$?

Q $\mathcal{D}(Y/X)$ → Distribution

Probability -

$Y = f(x) + \epsilon$

target function (T)

Q In reality most of the things follow normal estimate?

Q Which loss function we will use for which distribution (Suppose, normal distribution)

PAGE No. DATE: / 202

$\{f(x_i) = E[Y/x_i]\}$ Same

$Y_i \sim N(f(x), \beta)$ → variance; mean

$P(Y/X_i, w_i) \sim N(Y/w^T\phi(x), \beta)$

$f(x)$ — $w^T\phi(x)$ (Non linear); $w^T x$ (linear)

Suppose

L - training points

(x_1, y_1) (x_2, y_2) _ _ (x_L, y_L)

$P(Y_1/X_1, w, \beta) = N(Y_1/X_1, w, \beta)$ → Maximize

we need to maximize this. (Prob. of getting $Y_1 \_\_ Y_L$ given $x_1, \_\_ x_L$ respectively)

Independent samples.

$\prod_{i=1}^{L} N(Y_i/w^T\phi(x_i), \beta) \rightarrow P(Y_1, \_\_ Y_L / x_1, x_2, x_L)$

mean = μ; σ

Maximize

$= \prod_{i=1}^{L} \frac{1}{\sqrt{2\pi}\beta} \exp^{-\left(\frac{x - w^T\phi(x)}{2\beta}\right)^2}$

Maxi - x ≃ Max log x

$\rightarrow \text{Max} \log \prod_{i=1}^{L} \frac{1}{\sqrt{2\pi}\beta} \exp^{-\left(\frac{-x - w^T\phi(x)}{2\beta}\right)^2}$

$= \text{Max} \; \frac{L}{2}\log\beta - \frac{L}{2}\log(2\pi) - \beta \sum_{i=1}^{L} (Y_i - w^T\phi(x_i))^2$

$$\text{Max} \rightarrow \sum_{i=1}^{L} (Y_i - w^T \phi(x_i))^2$$

$$\underset{w}{\text{Min}} \rightarrow \sum_{i=1}^{L} (Y_i - \cdots (w^T \phi(x_i))^2$$

Normal dis → min least square loss function.

Drawback

Least square loss function - sensitive to outliers.

→ Error is generalized because of square term.

2022/02/15

least sqr. poly. reg with normalization.

$$\text{Loss}(L) \quad \min_{w,b} \frac{\lambda}{2} \|w\|_2 + \sum_{i=1}^{L} (Y_i^o - (w^T \phi(x_i) + b))^2$$

$$\frac{\partial L}{\partial w} = 0$$

$$\frac{\partial L}{\partial b} = 0$$

obtain

$$\begin{bmatrix} w \\ b \end{bmatrix} = (\phi^T \phi + \lambda I)^{-1} \phi^T Y$$

$\ln \lambda = -18$ → Better regularization.

$\ln \lambda = 0$ → straight line.

$\ln \lambda = -\infty \Rightarrow \lambda = 0$ (No regularization)

If we increase the value of lambda, then there is too much regularization.

Tuning - finding optimal value of λ so that there is no overfit.

As the value of λ increase, the typical magnitude of the coefficients get smaller.

$\ln \lambda = -\infty$

$\lambda = e^{-\infty}$

$\lambda = \frac{1}{e^{\infty}} = \frac{1}{\infty}$

$\Rightarrow \lambda = 0$

(No regularization)

$\ln \lambda = -18$

$\lambda = e^{-18}$

$\Rightarrow \lambda = \frac{1}{e^{18}}$

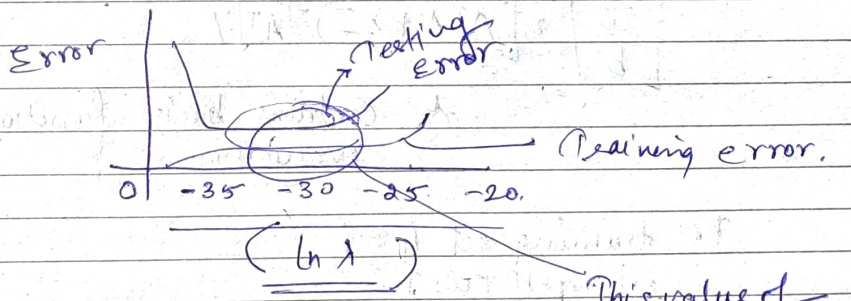
$\Rightarrow 1.52 \times 10^{-8}$

$\ln \lambda = 0$

$\lambda = e^0$

$\Rightarrow \lambda = 1$

(Straight line too much regularisation)



Now we will consider only this optimization problem.

$$\min_{w,b} \frac{\lambda}{2} \|w\|^2 + \sum_{i=1}^{L} (Y_i - (w^T \phi(x_i) + b))^2$$

Lab 2 Work

1) Plots of testing & training without regularisation.
2) Plots of —"— after regularisation.
   Print coeff. for this
   Print coeff. for this
3) Find optimal value of λ.

Tuesday
Date
17/02/2022

$$f(x) = w^T \phi(x) + b$$

we obtain this from training set.

$$\begin{bmatrix} w \\ b \end{bmatrix} = \left[A^T A + \lambda I\right)^{-1} A^T \Big] Y.$$

↳ A contains basis function evaluation.

For training set $T_1 = \{$
we get $\begin{bmatrix} w_1 \\ b_1 \end{bmatrix}$

These values after training with —"— P_2 —"— $\begin{bmatrix} w_2 \\ b_2 \end{bmatrix}$

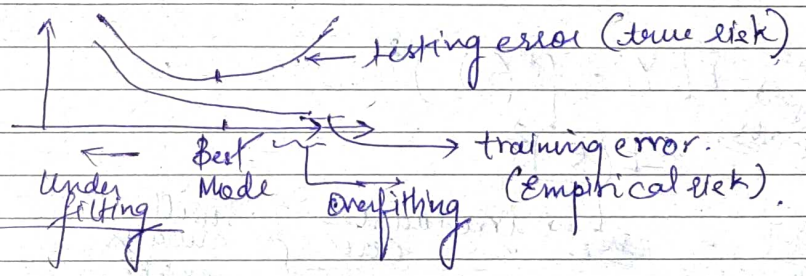
Non-linear
Page No. 2.5
DATE: / 202

In linear estimate, the variance is much less, than variance in non-linear estimate.

linear estimate	non-linear estimate
less variance	high variance
~~(High bias)~~	↳ bcoz non-linear model sensitive to outliers as we make more complex functions.

$F_1 \subset \cdots \subset F_8 \subset F_9 \subset F_{10}$

low variance → high variance.



- Modeling with fixed basis functions.

low deg Poly.	High deg.
High Bias (fits poorly)	Low Bias (fits too nicely)
low variance	High variance
↳ Underfitting	↳ Overfitting.

Avg. Sqr Error = (Bias Error)^2 + Variance

Actual Risk → should be estimated on entire dataset.

$\int (Y - f(x))^2 \, dP(x,y)$ → freq.

$E[(Y - f(x))^2]$

True Risk → on infinite data points

$R(f) = E[(f(x) - Y)^2]$

$= E[(f(x) - E[f(x)] - E$

Least Square Expected error $E[Y_i - f^*(x)]^2$

$= E[\epsilon]^2$

↳ irreducible error } will always exist.

$Y = f^*(x) + \epsilon$ ↳ actual relationship

Training set $T_D = \{x_1, y_1\}$

$f_{T_1}(x)$

$f_{T_2}(x)$

$E[f(x)]$

$E[(Y - f(x))^2]$ Error

↳ Estimated

$= E(Y - E_D(f(x)) + E_D(f(x)) - f(x))^2$

$= E(Y - E_D(f(x))^2 + E(E_D(f(x)) - f(x))^2$

$+ 2E[(Y - E_D(f(x)))(E_D(f(x)) - f(x))]$ (II)

Bias $\sin 2\pi(x_0) - (w_0 x + b)$

$\left(\frac{w_1 + w_2 + w_n}{n}\right)^T \phi(x) + \frac{(b_1 - \dots - b_2)}{n}$

(I) & (III) will turn to '0'

Term (II)

$2E\left[Y \cdot E(f(x)) - E(f(x)) \cdot E(f(x)) + f(x) \cdot Ef(x)\right)$ (constant, constant) $- Yf(x)$

$2\left[E\{Y E(f(x))\} - E\{E(f(x)) E(f(x))\}\right.$ (constant)

$+ E\{f(x) \cdot E(f(x))\}$

$\left. - E\{Yf(x)\}\right]$

$= 2\left[E(Y) E[f(x)] - (E[f(x)])^2 + (E[f(x)])^2 - E(Y) E[f(x)]\right]$

$= 0$

$E[Y/x] = f^*(x) = 0$

Variance, Bias

∴ LSE

↳ $E[Y - f(x)]^2 = E[(Y - E_D[f(x)])^2]$ (I)

$+ E[(E_D[f(x)] - f(x))^2]$ (II)

(I) ↳ $E_D[(Y - f^*(x))^2 + E_D[f^*(x) - E_D(f(x))]^2$

Irreducible Error + (II) Variance (Bias)

Highly non-linear models → High variance
Neural networks → low Bias
↳ But still perform better, — Why?
↳ Bcoz whenever we have large dataset, then variance is comparatively low

22/02/2022

linear $f(x) = w^T x + b \quad w \in R^n \quad b \in R \quad T = \{(x_1, y_1) \dots (x_2, y_2)\}$

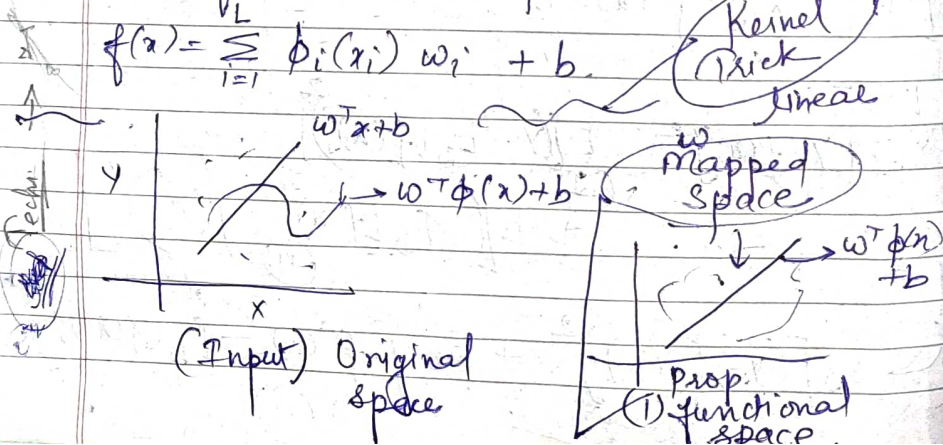
Non-linear $f(x) = w^T \phi(x) + b$.

$\phi(x) = [\phi_1(x) \ \phi_2(x) \dots \phi_M(x)]$ — Basis functions.

$\sum_{i=1}^{m} w_i \phi_i(x) + b$.

Suppose for every training point, we have a basis function.

$f(x) = \sum_{i=1}^{L} \phi_i(x_i) w_i + b$.



Higher dimensional Hilbert space → Infinite dimensional functional space

Mapped Space — $x_1 \longrightarrow [\phi_1(x_1)$

Kernel ~~Limits~~ Trick — Finding linear function in higher dimensional functional space is similar to finding non-linear func. in original space.

$K(x_1, x_2) = \phi(x_1)^T \phi(x_2)$ — Dot product in mapped space.

Kernel

analogous to reproducing Kernel space — Prop. (constraints) that K and ϕ should follow. Kernel mercer's conditions

Representing

$\phi(x) = \begin{bmatrix} \phi_1(x) \\ \phi_2(x) \\ \vdots \\ \phi_i(x) \end{bmatrix}$

$w = \sum_{i}^{L} u_i \phi(x_i)$ (w can be written as functional eval. of L training points)

$f(x) = w^T \phi(x) + b$

$= \left(\sum_{i=1}^{L} u_i \phi(x_i)\right)^T \phi(x) + b$

$= \sum_{i=1}^{L} u_i \cdot \phi(x_i)^T \phi(x) + b.$

$= \sum_{i=1}^{L} K(x_i, x) u_i + b$ → Kernel generated surface

we know

If, $K(x_i, x_j)$ know explicitly, then we do not need to explicitly $\phi(x_i)^T \phi(x_j)$.

Some popular Kernel

Kernel matrix → Symmetric Matrix

Kernel – should be the semidefinite ↳ Mercer condition.

x_i — $x_i \in R_n$.

$$\begin{bmatrix} K(x_1,x_1) & K(x_1,x_2) & \cdots & K(x_1,x_L) \\ & K(x_2,x_2) & & \\ K(x_L,x_1) & & & K(x_L,x_L) \end{bmatrix}$$

① Gaussian Kernel $\exp \frac{-\|x_1 - x_2\|_2}{\sigma}$

② $(x_1^T x_2)^c$

All kernels are Symmetric

Choice of kernel → is data dependent.

Eg RBF

$K(x_1, x_1) = 1$ (Both exactly same)

$K(x_2, x_1) = 0.2$ (More close)

$K(x_2, x_3) = 0.4$ (less close)

$f(x) =$

least-Squ. Kernel Regression ← we also apply regularisation.

$$\min_{u,b} \frac{\lambda}{2}\|u\|_2 + \sum_{i=1}^{L}\left(y_i - \left(\sum_{j=1}^{L} K(x_i, x_j) u_i + b\right)\right)^2$$

$\frac{dL}{du} = 0 \Rightarrow u = (X^T X)^{-1} X^T y$ ← random

here X is kernel matrix. with last col, are '1's

$$X = \begin{bmatrix} K(x_1,x_1) & \cdots & K(x_1,x_L) & 1 \\ K(x_2,x_2) & & & \\ & & & \\ K(x_L,x_1) & & K(x_L,x_L) & 1 \end{bmatrix}$$

A.K.a → least squ. support vector regression.

least Squa – is very sensitive to outliers.

$$J(w,b) \quad \min_{w,b} \frac{\lambda}{2}\|w\|^2 + \frac{1}{2}\sum_{i=1}^{L}\left(y_i - (w^T\phi(x_i) + b)\right)^2$$

→ Sol. of this optimization prob. using gradient descent algo.

$$\nabla_{w,b} J(w,b) = \lambda w + \sum_{i=1}^{L}(y_i - (w^T\phi(x_i) + b)) * (-\phi(x_i))$$

$x = w^T w = \|w\|^2$

$$\nabla_w w^T w = \begin{bmatrix} \frac{\partial f}{\partial w_1} \\ \vdots \\ \frac{\partial f}{\partial w_n} \end{bmatrix} = \begin{bmatrix} 2w_1 \\ 2w_2 \\ \vdots \\ 2w_n \end{bmatrix} = 2w$$

$$\nabla_w J(w,b) = \lambda w - \sum_{i=1}^{L}(y_i - w^T\phi(x_i) + b)\,\phi(x_i)$$

$$\nabla_b J(w,b) = 0 - \sum_{i=1}^{L} \left(y_i \left(w^T \phi(x_i) + b \right) \right)$$

$$x^{k+1} = x^k - \eta \nabla f(x^k)$$

η → data step length

Everything until $\| \nabla f(x^k) \|_2 \leq \epsilon$

Norm of gradient should be very very close to 0.

x^{k+1} → $\begin{bmatrix} w^{k+1} \\ b^{k+1} \end{bmatrix}$ = [.

Select some w & b

$\begin{bmatrix} 0 & 0 & 0 \end{bmatrix}$ $\begin{bmatrix} x_1 & x_2 & x_3 \end{bmatrix}$

b [.]

Evaluate.

x_1 — x_2

Reduce value of η is every iteration

$\eta(i) = \eta - \frac{\eta}{i}$

Better Convergence

Mid Sem Exam →

Generate Min point function A

Gradient descent

$A = \begin{bmatrix} & & \\ & & \\ & & \end{bmatrix}$ $\begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_m \\ b \end{bmatrix}$ $- \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_3 \end{bmatrix}$

$u = (A^T A + \lambda I)^{-1} A Y$ → $(Y - Au)$

In gradient descent $u = \begin{bmatrix} w_1 \\ \vdots \\ w_m \\ b \end{bmatrix}$ $\begin{bmatrix} w_1 \\ w_2 \end{bmatrix}$

Optimize

$$\min \frac{\lambda}{2} w^T w + \sum_{i=0}^{L} \left(y_i - (w^T \phi(x_i) + b) \right)^2$$

$$\min \frac{\lambda}{2} u^T u + (Y - Au)^T (Y - Au)$$ ← Matrix format.

$u^T I_0 u$ ← $w^T w$

When we add b^2 → stick to some frame of reference

$$J_\Delta(u) = \lambda u - A^T (Y - Au)$$

$$u^k = u - \eta \left(\lambda u - A^T (Y - Au) \right)$$

do this until $J_\Delta(u) <$ tolerance

Once done we move

gradient (λ, u, A, Y),

{ $x^2 + y^2 +$

} $x^2 +$.

line 162

M=1

2×20

x y

M=1

[x y 1]

M=2

[x² y² xy x y 1]

(M+1)

x² + y² + 3 + 2 + 1 = M+1

x²y⁰ x¹y¹ y²x⁰

$\frac{M(M+1)}{2}$

$= \frac{3 \times 4}{2} = 6$

$\binom{N}{1}$

$\binom{N}{2}$

$\binom{N}{M} = \frac{(M+1)}{2} \left(\frac{N(N+1)}{2}\right)$

24/02/2022

Suppose — $f(x) = ax + b$ good estimate.

Bcoz of some noise estimate deviated bcoz of outliers.

When we have least square loss estimate. it is very much prone to outliers.

How to overcome this?

Certain local measure — to calc. density of a point.

(M1) If this is too less, then we give less weight to that data point.

Dense points are given more weightage compared to outliers.

(M2) Distance from the predicted hyperplane

(M3) K-NN based..

Also we need a new loss function.

$$\min_{(w,b)} \frac{1}{2} w^T w + \frac{1}{2} \sum_{i=1}^{n} \| (Y_i - (w^T \phi(x_i) + b)) \|_1$$

L1 Norm Regression

L-1 norm → we are not squaring the error

$$\|X\|_1 = |x_1| + |x_2| + \dots + |x_n|$$

For kernel method.

$$\min_{(w,b)} \frac{\lambda}{2} w^T w + \underbrace{\sum_{i=1}^{L} |y_i - \left(\sum_{i=1}^{L} k(x_i, x_i) u_i + b\right)|}_{\text{Check}}$$

L_1 norm – (Not smooth at 0, Not differentiable)

L_2 norm ↳ (Always smooth & differentiable)

$$\min_{(w,b)} \frac{\lambda}{2} w^T w + \frac{1}{2} \sum_{i=1}^{L} (y_i - (w^T \phi(x_i) + b))^2 \quad \text{(1)}$$

(Most convenient)

Ridge

Rez. Regression → Min. L.S. loss along with regularization.

Lasso regression

$$\frac{\lambda}{2} \|w\|_1 + \frac{1}{2} \sum_{i=1}^{L} (y_i - (w^T \phi(x_i) + b))^2$$

* More simple exp. is guranteed to generalize well.

$$[w_1 \; 0 \; w_3 \; 0 \; 0 \; w_6] \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

Sparse sol.

Thus, variance of unwanted feature doesn't affect.

$$\min_{w,b} \frac{\lambda}{2} \|w\|_1 + \frac{1}{2} \sum_{i=1}^{L} |y_i - (w^T \phi(x_i) + b)| \quad \text{(2)}$$

L_1 norm of regular + L_1 norm of reg.

- for (1) we can use gradient descent.
- for (2) — we can't

Solve

$$\min_{w,b} \frac{\lambda}{2} w^T w + \frac{1}{2} \sum_{i=1}^{L} |y_i - (w^T \phi(x_i) + b)|$$

Use concept of sub-gradient.

Obtain gradient of

$$\nabla_w \left(y_j - (w^T \phi(x_j) + b)\right) = -1 \cdot \phi(x_j)$$

$$= \begin{cases} \nabla \frac{|z|}{z} \frac{z}{w} & \text{if } y_j - (w^T \phi(x) + b) > 0 \\ -1 \times -1 \times \phi(x_i) & \text{if } (y_j - (w^T \phi(x) + b) < 0 \end{cases}$$

$$= \phi(x_i)$$

Check for every training point.

We have to calc. gradient for every sample point individually

Code L_1-norm regression.

Similarly also for.

$$\left| y_j - \left(\sum_{i=1}^{L} K(x_i, x_j) u_i + b \right) \right|_\epsilon$$

Vector form $\nabla = \begin{bmatrix} u \\ b \end{bmatrix}$ $\quad u = \begin{bmatrix} w \\ b \end{bmatrix}$

$Y - Ru$

Diff. w.r.t u_i

$$= \begin{cases} -\sum_{i=1}^{L} K(x_i, x_j) & \text{if } y_j - \left(\sum_{i=1}^{L} K(x_i, x_j) + b \right) > 0 \\ \sum_{i=1}^{L} K(x_i, x_j) & \text{if } y_j - \left(\sum_{i=1}^{L} K(x_i, x_j) + b \right) > 0 \end{cases}$$

How to select the value of step length η

Laplace Reg. Model (we just solved)

$$\frac{1}{2} w^T w + \frac{1}{2} \sum_{i=1}^{L} \left| y_i - \sum_{j=1}^{L} K(x_j, x_i) u_j + b \right|$$

Ask a Laplace Support Vector Reg. Model.

L_1-norm loss also called as laplace loss.

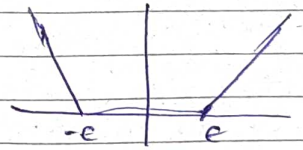
$L(u) = u^2$ least square loss with L_1 norm regularization or (L_2 norm loss) and L_2 norm regularization

$L(u) = |u|$ → L_1 norm loss ——— " ———

" "

ϵ - insensitive loss function.

$$L_\epsilon(u) = \begin{cases} 0 & \text{if } |u| < \epsilon \\ |u| - \epsilon & \text{otherwise.} \end{cases}$$



beyond ϵ, it becomes L_1 norm loss

Kernel Matrix (without b)

$$\min_{w,b} \frac{\lambda}{2} w^T w + \frac{1}{2} \sum_{i=1}^{L} L_\epsilon \left(Y_i - \sum_{j=1}^{L} K(x_j, x) u_j + b \right)$$

Thus, $L_\epsilon(Y_k - \sum_{j=1}^{L} K(x_j - x) u_j + b)$ $\rightarrow (w^T \phi(x_j) + b)$

can be given as

$$= \begin{cases} 0 \\ Y_k - \sum_{j=1}^{L} K(x_i, x_k) u_j + b) - \epsilon \end{cases}$$ otherwise

© Suppose if we are using linear kernel.

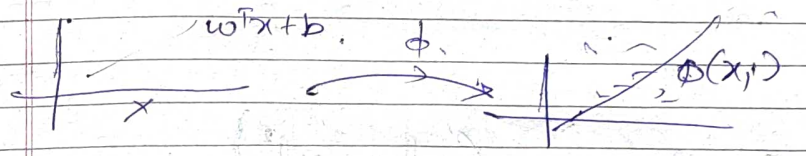
$$\min_{w,b} \frac{\lambda}{2} w^T w + \frac{1}{L} \sum_{i=1}^{L} L_\epsilon (Y_i - (w^T x + b))$$

$$L_\epsilon (Y_k - (w^T x_k + b))$$

$$= \begin{cases} 0 \\ |Y_k - (w^T x_k + b)| - \epsilon, \text{ otherwise.} \end{cases}$$

Suppose we have L basis function.

$w^T \phi(x_j) + b$ $\quad \begin{bmatrix} \phi_1(x_j) \\ \phi_2(x_j) \end{bmatrix}$



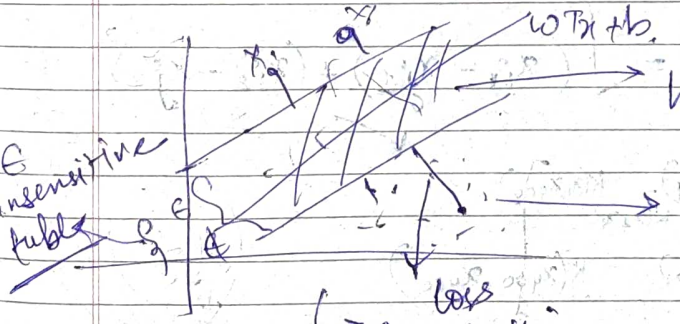
Representer Thm
" w can be written as linear comb. of L training points "

$$\sum_{i=1}^{L} \phi(x_i)^T \phi(x_j) u_i + b$$

$$= K(x_j, x_j)$$

In support

$$\sum_{i=1}^{L} K(x_j, x_i) u_i + b$$



here error is 0.

∴ Corresponding point have 0 loss assigned.

↳ Good thing — Solution is going to be sparse.

$$\min_{w,b} \frac{1}{2} w^T w + C \sum_{i=1}^{L} \left| y_i \left(w^T \phi(x_i) + b \right) \right|_{\epsilon}$$

$$U = (K + \lambda I)^{-1} \cdot Y$$

K → size L × L

Q1

$$K(x,y) = \exp\left(- \frac{\| x - y \|^2}{\sigma} \right)$$

Suppose we have

$(x_1, y_1 \; \ldots \; x_{400}, y_{400})$

and each $x_i^\circ = (x_{i1}, x_{i2})$

$$K(x_i^\circ, x_j^\circ) = \exp\left(- \frac{\| x_i^\circ - x_j^\circ \|^2}{\sigma} \right)$$

and

$$\| x_i^\circ - x_j^\circ \| = \sqrt{(x_{i1} - x_{j1})^2 + (x_{i2} - x_{j2})^2}$$

$$K = \begin{bmatrix} K(x_1, x_1) & \cdots & K(x_1, x_{400}) \\ \vdots & K(x_2, x_2) & \\ K(x_{400}, x_1) & & K(x_{400}, x_{400}) \end{bmatrix}$$

$A = \phi(x)$ = [ ] $L \times$ Col ordered M

A^T = [ ] Col × L

Col $= \frac{(M+1)(M+2)}{2}$

$A^T \cdot A =$ Col × Col

eg: M = 5

Cols $= \frac{6 \times 7}{2} = 21$

Without Regularisation.

$$U = (A^T \cdot A)^{-1} \cdot A^T \cdot Y$$

With Regularisation

$$U = \left((A^T \cdot A) + \lambda I \right)^{-1} \cdot A^T \cdot Y$$

$Y_{pred} = A \times U$

$(L \times (M+1)) \cdot ((M+1) \times 1) = (L \times 1)$

K – Kernel Matrix

L Data points

K → size L × L

Since K does not have 1's in last col for bias term, we will make $K = [K \; 1]$

$$U = (K^T K)^{-1} K^T Y$$

$$U = (R^T \cdot R + \lambda I)^{-1} R^T Y$$

part 1

R → L × L+1 rows cols.

$Y_{pred} = R \cdot U$

$(L \times (L+1)) \cdot ((L+1) \times 1) = (L \times 1)$

$R^T \times R$

$(L+1) \times (L+1)$

$I \rightarrow (L+1) \times (L+1)$

↳ identity matrix.

final size of U

part 1 R^T $(L+1) \times L$

$(L+1) \times L+1$

$(L+1) \times 1$

$U =$ [ ] $(L+1) \times 1$

$(L+1) \times L$ · Y

$L \times 1$

DATE: / 202

Observation

Trained

	RMSE
$\sigma = 1$	0.4
$\sigma = 0.05$	0.27
$\sigma = 0.01$	0.28

high value of σ, higher value of RMSE.
Poor predictions.

Test set

(21)

401 x 1 → Coefficients

K_test = [K(x_test_i , x_train_j)]
↳ size 200 x 401

Y_test_pred = K_test * U.
[200 x 1]   [200 x 401]   [401 x 1]

Y_pred   Kxd

Q2

Q3

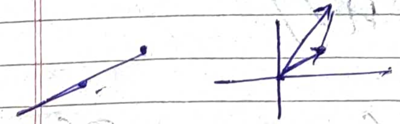
PAGE No
DATE: / 202

0, 1, 2, 3, 4, 5

$x_1 - x$ (1)

gradient

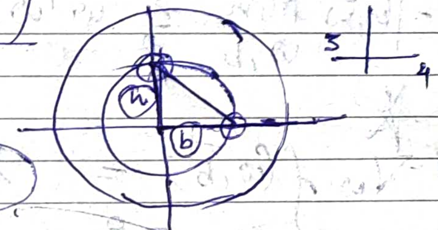
$\sqrt{3^2+4^2}$



20 marks
PRML
1 hour

Mid term exam
→ Not many numerical

$$G_1 = \lambda U - A^T(Y - AU)$$



x_i

L1

A (p, q)

B (x, y)

$\sqrt{(p-x)^2 + (q-y)^2}$

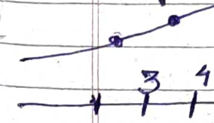
A (p)

B (x)

$\sqrt{(p-x)^2} = (p-x)$ → L1 norm

(3) (4) (5)

3 — 4



3 4 $\sqrt{3^2+4^2}$

0 2 3 4

$w^T x + b$

$\lambda U - A^T \cdot (Y - AU)$

just replace with kernel

$\text{Grad} = \lambda V + A^T (Y - HV)$

$v^{k+1} = v^k - \eta \left(\lambda v + H^T (Y - Hv) \right)$

L_1 - norm loss

$$f(w,b) \quad \min_{w,b} \frac{\lambda}{2} w^T w + \sum_{i=1}^{L} \left| (Y_i - (w^T x_i + b)) \right|$$

$$\nabla_w f(w,b) = \lambda w + \begin{cases} -x_i & \text{if } (Y_i - (w^T x_i + b)) > 0 \\ x_i, & \text{otherwise} \end{cases}$$

$\frac{\partial f(u)}{\partial u} = 1$ if $u > 0$, otherwise

$$f(w_1, w_2, b) = \min_{(w_1, w_2, b)} \frac{\lambda}{2} (w_1^2 + w_2^2 + b^2) + \sum_{i=1}^{L} \left| (Y_i - (w_1 x_1 + w_2 x_2 + b)) \right|$$

L

$$\frac{\partial L}{\partial w_1} = \lambda w_1 + \sum_{i=1}^{L} \begin{cases} -x_1 & \text{if } (Y_i - (w_1 x_1 + w_2 x_2 + \dots) > 0 \\ x_1 & \text{otherwise} \end{cases}$$

$$\frac{\partial L}{\partial w_2} = \lambda w_2 + \sum_{i=1}^{L} \begin{cases} -x_2 \\ x_2 \end{cases}$$

$$\frac{\partial L}{\partial b} = \lambda b + \begin{cases} -1, \\ 1 \end{cases}$$

$$\nabla_b f(w,b) = \lambda b + \begin{cases} -1 & \text{if } (Y_i - (w^T x_i + b)) > 0 \\ 1, & \text{otherwise} \end{cases}$$

$Y = AU$

Vector form

$$= \frac{\lambda}{2} u^T u + \| Y - Hu \|_1$$

$$u - H_i \quad \text{if } Y - Hu > 0$$

$u + H$

$$\text{Grad} = U + \begin{cases} -H_i^T & \text{if } Y - HU_i < 0 \\ H_i^T & \text{otherwise} \end{cases}$$

$$H = \begin{bmatrix} x_{11} & x_{12} & x_{1m} \\ x_{21} & x_{22} & \end{bmatrix}$$

$Y_i - H_i(i) > 0$

$H(i)\,u$

$$Hu = \begin{bmatrix} w^T x_1 + b \\ w^T x_2 + b \\ \vdots \\ w^T x_L + b \end{bmatrix}$$

$$Hu(i) = w^T x_i + b$$

$$u - \eta \rightarrow \quad Y - (w^T x_i + b) > 0$$

$$Y - Hu(i) > 0$$

$$u^{k+1} = u^k - \eta$$

$$-\eta \; u^k + \begin{cases} -H_i^T, & Y_i - Hu^k(i) > 0 \\ H_i^T & \text{otherwise} \end{cases}$$

Gradient $U^{k+1}_{S1} - \theta_{S1}$ [ ] Return U

PAGE No
DATE: / 202

(Mid Sem - 25th)

Extreme ML → Activation

x_1 → $g(w_1^T x + b_1)$

x_2 → $g(w_2^T x + b_2)$

$$\hat{Y}_i = \sum_{j=1}^{L} g(w_j^T x_i + b)\beta_j$$

Estimate $= g(w_1^T x + b)\beta_1 + g(w_2^T x + b_2)\beta_2 + \ldots$

Minimize least square:

$$\min_{\substack{w_1, w_2, \ldots w_L, b \\ \beta_1, \beta_2 \ldots \beta_L}} \sum_{k=1}^{L} \left(Y_k - \sum_{j=1}^{L} g(w_j^T x_k + b)\beta_j \right)^2$$

Just minimize this $\beta_1, \beta_2 \ldots \beta_L$, to get the optimal solution

→ Only this also suffice

$$\underset{(\beta_1,\beta_2,\ldots\beta_L)}{\text{Min}} \frac{1}{2} \sum_{k=1}^{N} \left(Y_k - \sum_{j=1}^{L} G(w_j^T x_k + b_j)\beta_j \right)$$

$$Y = \begin{bmatrix} Y_1 \\ \vdots \\ Y_N \end{bmatrix} \qquad H = \begin{bmatrix} G(w_1^T x_1 + b) & G(w_2^T x_1 + b) & \cdots & G(w_L^T x_1 + b) \\ \vdots & & & \\ G(w_1^T x_N + b) & & & \end{bmatrix}$$

N = # of data points.

L = no. of hidden nodes / weights

$N \times L$ size.

$$\beta = \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_L \end{bmatrix}$$

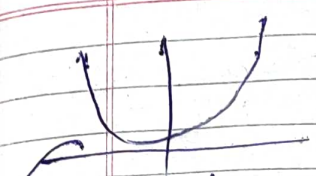
$$\underset{\beta}{\text{Min}} \left\{ \frac{1}{2} (Y - H\beta)^T (Y - H\beta) \right\}$$

$$\therefore \left\{ \beta = (H^T H)^{-1} H^T Y \right\}$$

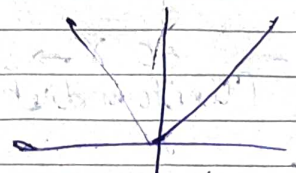
$(+\lambda I)$ for regularisation

↳ Only this is done in extreme machine learning.

Recall $A = \begin{bmatrix} \phi_1(x_1) & \cdots & \phi_L(x_1) \\ \phi(x_N) & & \phi_L(x) \end{bmatrix}$



↳ when noise follows $\sim N(\mu, \sigma)$



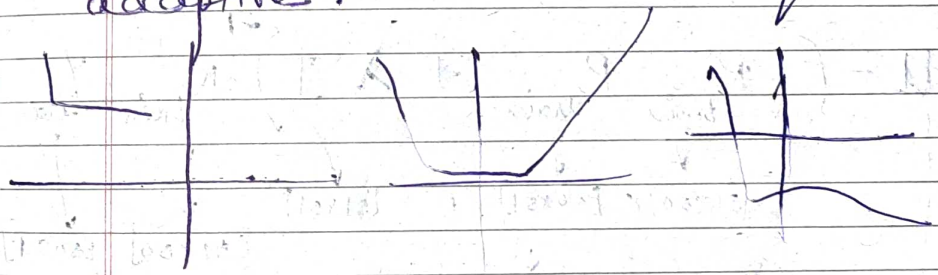
↳ when noise follow uniform

$e^{-|z|}$

∴ We need to find distribution of noise.

↳ Which is not easy.

Thus, we need to make loss function adaptive.



Piece-wise

Arrin

learning a powerful SVM with piecewise linear loss function"

Tuesday

Assignment-4

(1) Univariate – 50 X_{train} ('Uniform dist^n').

(2) $Y_{train} \rightarrow \sin(2\pi x_i) + e_i$

(3) 500 - testing points – (X_{test}, Y_{test})

(4) Regularised least square kernel regression

Kernel RBF

$$K_{train} = \begin{bmatrix} & \\ & \end{bmatrix}_{50 \times 50}$$

$$R_{train} = \begin{bmatrix} K_{train} & 1 \end{bmatrix}_{50 \times 51}$$

$$U = \left(R_{train}^{T} R_{train} + \lambda I \right)^{-1} R_{train}^{T} \cdot Y_{train}$$

$([51 \times 50] \times [50 \times 51] + [51 \times 51])$

$[51 \times 50]$ $[50 \times 1]$

$[51 \times 51]$

$[51 \times 1]$ (dimension)

(Training estimate) $\hat{Y}_{train\text{-}pred} = R_{train} * U$

$[50 \times 51]$ $[51 \times 1]$

$[50 \times 1]$

For test RMSE.

$$\hat{Y}_{test\text{-}pred} = R_{test} * U$$

$[500 \times 51]$ $[51 \times 1]$

where

$$R_{test} = \begin{bmatrix} K_{test} & 1 \end{bmatrix}$$

$$K_{test} = \begin{bmatrix} & \\ & \end{bmatrix}_{500 \times 50}$$

and

$$K_{test}(i,j) = \exp\left(- \frac{\| x_i - x_j \|}{\sigma} \right)^2$$

where $x_i \in X_{test}$

and $x_j \in X_{train}$

Plot $(Y_{test}, \hat{Y}_{train\text{-}pred}, \hat{Y}_{test\text{-}pred})$.

Q5 $X_{train\text{-}modified}$ (Has some outliers) → Select 5 training points & modify their $Y_{i\text{-}mod} = Y_i + 4$

Q6 Performance comp. Q4 & Q5.

Q7 L1-norm loss kernel Regression Model.

$R_{train} \rightarrow [50 \times 5]$

Gradient descent.

$U_0 = \begin{bmatrix} \vdots \\ 1 \end{bmatrix}_{5 \times 1}$

$U_{k+1} = U_k - \eta (Grad_{vec})$

grad_vec = gradient ( )

gradient (Y_{train}, R_{train}, U)

[As per class doubt session] $R_U = R_{train} * U$ (50×1)

grad_vec ??? → (i - ?) how to select)

$\rightarrow U + \begin{cases} -R^T_{train}(i) & \text{if } (Y_{train}(i) - R_U(i)) > 0 \\ +R^T_{train}(i) & \text{otherwise} \end{cases}$ ($i \rightarrow [5 \times 1]$)

→ Since we have to consider every training point, how we will select (i) ?

→ Can this missing area be replaced with $\sum_{i=1}^{i=50}$?

$\Delta Y = Y_{train}(i) - R_U(i)$ | $R_U = R_{train} * U$

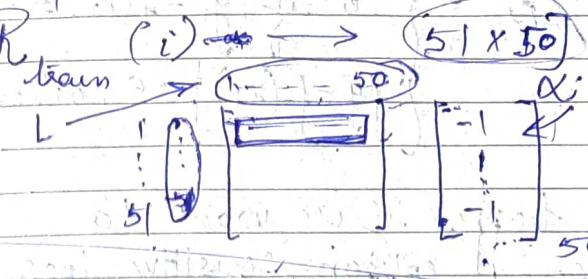
$i \rightarrow 1, 2, 3, \ldots, 50$

$\Delta Y \rightarrow [1 \; \ldots]$

If $\Delta Y_i > 0$, $\alpha_i = -1$ | else $\alpha_i = 0.1$

$\alpha_{vec} = [-1 \quad 1 \quad \ldots \quad -1]^T_{50 \times 1}$

$R^T_{train}(i) \rightarrow [5 \times 50]$



$grad_{vec} = U + \varepsilon^T R^T_{train} \alpha_i$

$\rightarrow U + R^T_{train} \cdot \alpha$

23/03/2022

Epilson insensitive loss function.

$$L_\epsilon(u) = \begin{cases} 0 & \text{if } |u| \le \epsilon \\ |u| - \epsilon, & \text{otherwise} \end{cases}$$

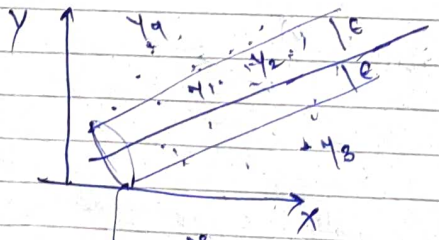
$u = y_i - f(x_i)$

$$L_\epsilon(y - f(x_i)) = \begin{cases} 0 & \text{if } |y_i - f(x_i)| \le \epsilon \\ |y - f(x_i)| - \epsilon, & \text{otherwise} \end{cases}$$

What will happen if we want to minimize the epilson sensitive loss function

$f(x) = w^T x + b$.

$w \in R$ and $b \in R$



for y_1 and y_2 loss will be 0

for y_3 & y_4 loss will be -tve number

Data points lying inside epilson insensitive tube will not contribute to error

Parametric methods — Bayesian methods

Statistician → maximum likelihood methods.

Fisher discriminant Analysis.

Neural networks bcoz assumption on distribution were not working.

Parametric methods - assume data is following some distribution (& we estimate parameters for it)

Unknown parametric - data distribution is not known but distribution will be fixed.

Book — Duda Hertz — Chap1 & Chap2 - Classification.

Fish classification problem.

Class - $(w_1 \quad w_2)$.

$X = (x_1, x_2)$

Given w_1 class, what is the distribution of fishes.

→ (X/w_1)

" - w_2 - " "

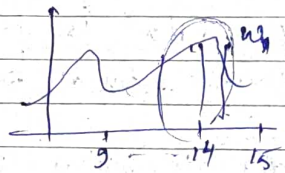
→ (X/w_2)

Given feature X what is the prob. fish belongs to w_1 class.
$P(w_1/X)$.

——— ——— ——— w_2 class
$P(w_2/X)$.

Prob. of fish being w_1 → $P(w_1)$.
w_2 $P(w_2)$

Density function of distribution of w_1 class
↳ How much dense our points in particular region.



If density function is higher at some point, that means lot of data points are accumulated.
Eg: length → 14

Bayes formula

$$P(w_j|x) = \frac{p(x|w_j)\,P(w_j)}{P(x)}$$

posterior = $\frac{\text{likelihood} \times \text{prior}}{\text{evidence}}$

$$p(x) = \sum_{j=1}^{2} p(x|w_j)\,P(w_j)$$

distribution of x in w_1 class called likelihood

$$P(w_1|x) = \frac{P(x|w_1) \times P(w_1)}{P(x)}$$

Prior

$P(x)$ ↓ evidence.

$$P(x) = P(x|w_1)\,P(w_1) + P(x|w_2)\,P(w_2).$$

Binary classif.: Decide w_1 if $P(w_1|x) > P(w_2|x)$
otherwise w_2.

Multi class classif.: Decide w_j if $P(w_j|x) \geq P(w_i|x)$ $\forall i = 1, 2, \ldots k$ and $i \neq j$

Error in binary classification.

$$P(w_1/x) + P(w_2|x) = 1.$$

$$P(\text{error}|x) = \begin{cases} P(w_1|x) & \text{if we decide } w_2 \\ P(w_2|x) & \text{if we decide } w_1 \end{cases}$$

Minimize error. (the avg. prob. of error).

$$P(\text{error}) = \int_{-\infty}^{\infty} P(\text{error}|x)\,P(x)\,dx.$$

↳ This integral will result in smaller error.

dependent variable K dependent on x.

$$P(K) = P(K|x) \times P(x).$$

$$P(K) = \int_{-\infty}^{\infty} f(K|x)\,f(x).$$

PRML

5:30

1hr exam — Mid Sem —

also $P(\text{error}|x) = \min\left(P(\omega_1|x), P(\omega_2|x)\right)$

$P(\omega_1|x) > P(\omega_2|x) \Rightarrow \omega_1$ class, otherwise ω_2 class → Bayesian decision Rule

$\Rightarrow \frac{P(x|\omega_1)P(\omega_1)}{P(x)} \geq \frac{P(x|\omega_2)P(\omega_2)}{P(x)}$

** Statistically most powerful (But data should be known)

Decide: ω_1 : if $P(x|\omega_1)P(\omega_1) \geq P(x|\omega_2)P(\omega_2)$

New decision criteria. ← Thus, likelihood & prior prob factors are important

here evidence can be omitted.

Multiclass classification

↳ Concept of loss function.

↳ But usually entire universe of data is not known.