Machine learning:
Statistical learning

$S_n = \{(x_1, y_1), \dots (x_n, y_n)\}$ · training set

$x_i \in X \subseteq \mathbb{R}^\alpha$

$y_i \in Y \rightarrow$ Scalar Regression $\rightarrow$ Vectorial Regression
$Y \subseteq \mathbb{R}$ $Y \subseteq \mathbb{R}^m$

Binary classification $Y = \{-1, 1\}$ $0, 1$

Multi-class classification (C)
$Y = \{0, \dots C-1\} \leftarrow$ ordine

I want my function to fit & generalize new data

$f: X \rightarrow Y$

$f(x_i) \simeq y_i$ fitting

$f(x_{new}) \simeq y_{new}$ $x_{new} \in S_n$ generalization

in training set we have not x_{new} but function must work

f must have good approximation.

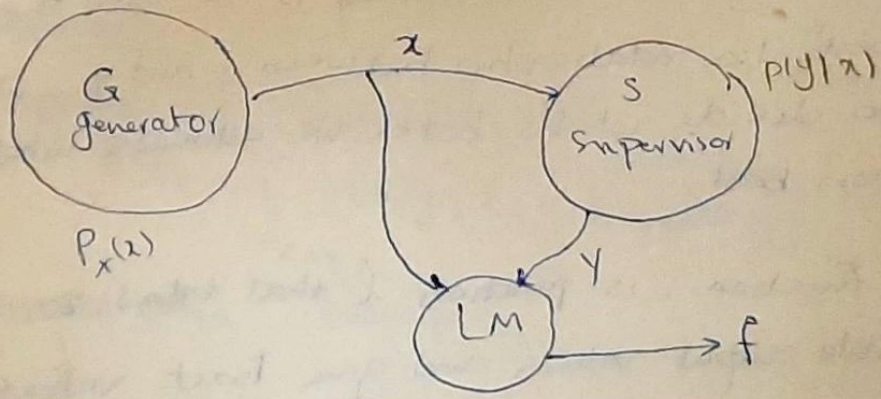
Ass 1: input part of training set generation of probability distribution of X such that the samples

x_i: one i.id

x_i: sample according to probability set has been identically and independently sample (i.id)

Ass 2: ~~input part of training set~~

$p(x,y) = P_x(x) p(y|x)$



What we looking for f: strongly related to:

$p(x,y) = P_x(x) p(y|x)$

generative model try to know probability of

$p(x,y) = P_x(x) p(y|x)$

in Supervised learning we want to estimate function not Probibility distribution.

Dont forget:

$X \times Y$ is our data space.

for anoment:

function f is the best goal

let's say: Goal of learning is to find the "best" I/O relationship between x and y

but to decide what's better we estimate what solution best.

Loss Function: is function l that takes 2 Possible output values and give back values $[0, +\infty)$, tells you:

$l : Y \times Y \longrightarrow [0, +\infty)$

$l(y_i, f(x_i))$ Common (popular choice)

$x_i \rightarrow y_i$

$\rightarrow f(x_i)$ if $y_i = f(x_i) \rightarrow$ perfect

but we have difference.

$\|y_i - f(x_i)\|^2$: norm, square loss

if these 2 function are equal prediction and answer would be the same and this is a goal.

but in reality this difference is and this is equal to norm & regression

Defination:

Expected loss (expected risk)

$$\mathcal{E}(f) = \int_{X \times Y} p(x,y)\, l(y, f(x))\, dx\, dy$$

expected value ↓ $\mathbb{E}_{XY}$ ($l(y, f(x))$ ← for prediction of other value which is added to the set

Expected value on all data space is add x to past & future.

The best I/O function $(f^*: X \to Y)$ ✓

$f^* = \text{argmin}\ \varepsilon(f)$

$f \in \mathcal{H} \quad f \in Y$ → whatever type of f.

I want to find min $\varepsilon(f)$

arg min of f associated with min function

***can not be calculated.

Problem:

1, we need probibility

if probibily change everything must be changed

2, we don't have access to entire dataspace

what we have a training set with sample

we need function depends on (n)

if n change the solution changed

for solve:

Empirical loss / Risk

$$\hat{\varepsilon}_n(f) = \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, f(x_i))$$

↑ sum

Limited number of sample so we can compute it

$\hat{f} = \text{arg min}\ \hat{\varepsilon}_n(f) \quad f \in \mathcal{H}$ → Hypothesis space

↓

Best I/O relationship, best function ✓

$\hat{f}: X \to Y \quad X \subseteq \mathbb{R}$

$Y \subseteq \mathbb{R}$

$\mathcal{H}$ $f(x) = wx + b$ → class of linear functions we call $\mathcal{H}_1$

$\mathcal{H}_1$: { poor fitting / High stability

$\mathcal{H}_2$: very complex function

$\mathcal{H}_2$: { High fitting / not stable (poor stability)

if I change 1 point it changes alot. ✓

overfitting . H_2 ...

not stability ... fit

finding best I/o

...

Finding the best I/o relationship from the data which means learning from examples means to find a compromise between fitting and stability generalized.

$\mathcal{E}(f^*)$ $\mathcal{E}(\hat{f})$ → estimated from data

↓ ideal quantity

2 different function.

if I have 2 above function, as I increase samples, difference of $\mathcal{E}(f^*)$, $\mathcal{E}(\hat{f})$ is going to Zero.

Consistancy:

$\lim_{n \to +\infty} \mathbb{E}\left(\mathcal{E}(f^*) - \mathcal{E}(\hat{f})\right) = 0$ ← zero ✓

number of samples.

— session 3 - 9th Oct

local method Nearest Neighbour [5]

I only consider 1 or efew point to take decision

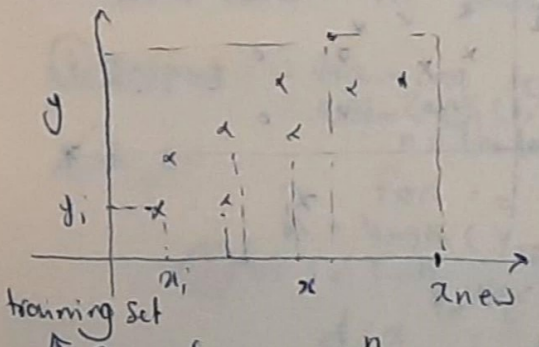
start from assumption: find estimator $\hat{f}$

$\hat{f}: X \to Y$

if you have 2 in input space should close in output space.

How explode:

example: talk about function $\hat{f}: X \to Y$, $X \subseteq \mathbb{R}$, $Y \subseteq \mathbb{R}$



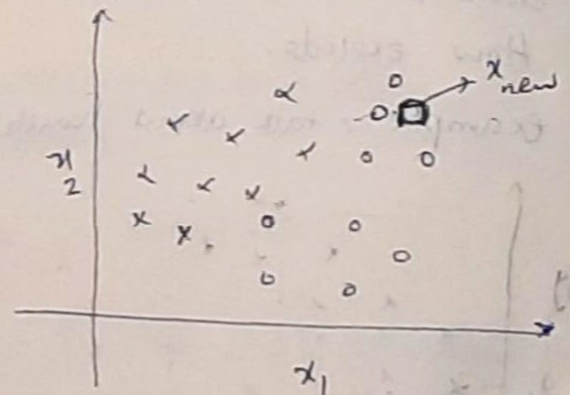
$S_n = \{(x_i, y_i)\}_{i=1}^{n}$

↓ number of element = 9 in example

lets assume : give a new point (x_{new})
ask estimator what are the act of it
for new point:

$\hat{f}(x_{new}) = ?$

I decide output of x_{new}
what we I have Binary classification
2 class x and o
2 cordinate,
instead of having x and y



$\hat{f}: X \rightarrow Y$
$\subseteq \mathbb{R}^2$ $\in \{x, o\}$

برای تعیین x_{new} در این حالت نزدیک ترین همسایه را پیدا می‌کنیم سپس x_{new} را همسایه‌ی
که کمترین فاصله را با آن دارد در نظر می‌گیریم.

فاصله
$\| x_{new} - x_i \|$
$i = 1, \dots n$ $\Rightarrow$ بردار x_i که کمترین فاصله را با x_{new} دارد پیدا کنیم

$$i^* = \arg\min_{i=1,\dots n} \| x_{new} - x_i \|$$

$$y_{new} = \hat{f}(x_{new}) = y_{i^*}$$

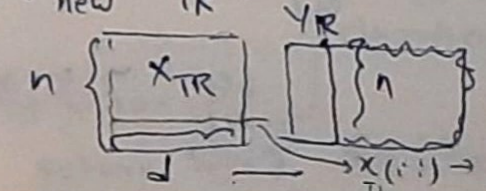
اگر این روش به اندازه‌ی کافی دقیق نباشد می‌توان از k همسایه‌ی نزدیک استفاده کرد و رای‌گیری انجام داد.

در این شرایط explicit داریم

Let's turn to sudocode

① INput : training set X_{TR} Y_{TR} and x_{new}
Dist = Zeros (1, n)
n = the length of training set (x, Y)
for i = 1 ... n
n = length (Y_{TR})

② for i = 1 ... n
$d(x_{new}, X_{TR}(i, :))$



Nearest neighbours

Input: X_{TR} Y_{TR} : traing set

x_{new}

Dist = Zeros (1, n)

n = Length (Y_{TR})

for $i = 1 \ldots n$

↳ $Dist(i) = d(x_{new}, X_{TR}(i, :))$

(minval, minidx) = min (DIST)

return Y_{TR} (minidx)

Computation of Complexity of method.

$O(nd)$

↙ all samples of TR

↑ → could be high dimention

K nearest neighbours

=

K is a number of closes point to take position

choise of K which are better than others

K : hyper parameter

outside the method.

if K is changed get another distance of method

INput : X_{TR} , Y_{TR} , x_{new} , K → is a number

if K is 1 : algorithm is nearest neighbour

n = lenght (Y_{TR})

we need Dist : Zero (1, n)

for $i = 1 \ldots n$

$Dist(i) = d(x_{new}, X_{TR}(i, :))$ row

(1) S_Dist = Sort (DIST, 'ASCEND')

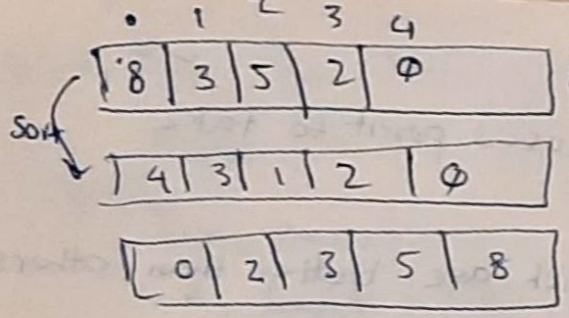
(1) (S_DIST, S_IND) = Sort (DIST, 'ASCEND')

(II) Regression

return $\frac{1}{K} \sum_{j=1}^{K} Y_{TR}(S_{IDX}(j))$

(II) Binary classification

return $\text{Sign} \sum_{i=1}^{K} Y_{TR}(S_{IDX}(j))$



; S_IDx

intrest the first key indeces give me

I have k possible inDx

if I have regresson pgblem: do avarage

classification:

sum = ϕ , might have {-1, 1} on boundaries

K must be odd number

half vote for +1

" " -1

So K must be odd

$O(nd + n\log n)$ ← sort

if change k, algorithm to predict output changes.

if try and classify with k=1

end up having: (KNN classify) in python notebook

خودش برای تمرین فعالش کرده

x+e

---

session: 14 Oct 2024

$(x_i, y_i)_{i=1}^{m} \rightsquigarrow \hat{f} : X \to Y$

$f(x_{new}) \sim y_{new}$

$X \in \mathbb{R}^d \quad x = (x^1 \dots x^d)$: collection of number

↑ Component of vector

$Y \subseteq \mathbb{R} \quad Y = \{\pm 1\}$ ← حالت خاص → برای رگرسیون Y عضو مجموعه دلخواه

$(x_i, y_i)_{i=1}^{m} \sim P^m$ i.i.d

① $P(X, Y)$ prob distribution on (X, Y)

all data are identical & independent

$= P_x(x)\, p(y|x)$

② $\ell : \mathbb{R} \times \mathbb{R} \to \mathbb{R}_+$, $\ell(f(x), y)$

$\min \mathcal{E}(f)$ $\quad \mathcal{E}(f) \triangleq \mathbb{E}_{(x,y)\sim p} [\ell(f(x), y)]$

$f : X \to Y$ ↓ expected error future data

$S_n \sim p^m$

given $S_n \sim p^m$

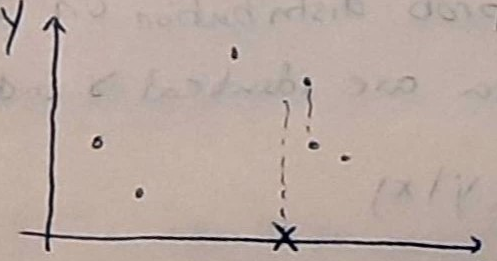
K-NN

$f(x) = \sum_{i=I_x} y_i / k$ میانگین

$I_{x,k} \subset \{1 \dots m\}$; $d_i = \|x - x_i\|^2$ $\quad i = 1 \dots m$

$(\tilde{I}, \tilde{d}) = \text{Sort}(d_1 \dots d_m)$

$I_{x,k} = \tilde{I}(1 \dots k)$

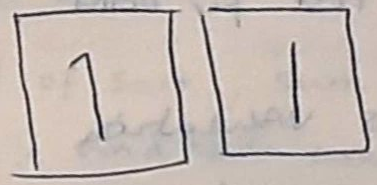


y که متناسب با هم جمع می‌شوند برای y_{new}

① $\hat{Y} = \text{K-NN}(\hat{X}, \hat{Y}, X, k)$
$m \times d$ $m \times 1$

close in input, so close output

با این تقسیم‌بندی کار

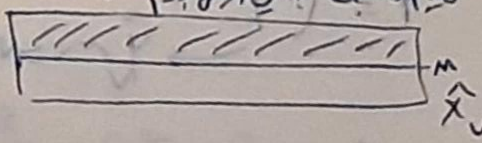


شکل pixel نگاه کنید
هم تفاوت دارند

هستیم نحوه این دو تا را بر حسب distance مرتب خواهیم کرد

و بعد هم با این فاصله را مرتب می‌کنیم.

$(\hat{x}, \hat{y}) = (x_i, y_i)_{i=1}^{m}$
$m \times d$, $m \times 1$



$\hat{y} = \begin{pmatrix} \to d \\ m \\ \sim \end{pmatrix} \hat{y}_v$

② $\min \|\hat{y}_v - \hat{y}_{v,k}\|^2$ $\quad \hat{k} \to \hat{\mathcal{E}}_{cv}(k)$

با تقسیم داده از ابتدا شروع می‌کنیم
با داده‌های train شروع می‌کنیم
از آن‌ها برای test استفاده می‌کنیم

Take Data

half for function
try for find K
an half for test for better

Hold-Out CROSS VALidation
↳ Hold apart of data out

X_{val} → validation set.

$\hat{Y} = (\ ) \to \hat{Y}_T \rightsquigarrow T$

$\to \hat{Y}_V \rightsquigarrow$ validation

split random

$\hat{Y}^q_{V,K} = K\text{-}NN(\hat{X}^q_T, \hat{Y}^q_1, \hat{X}^q_V, K)$

$\hat{Y}^q_V - \hat{Y}^q_{V,K} = \hat{\varepsilon}_{V,q}(K)$

K=1	2	
$\hat{\varepsilon}_{V,1}(1)$	$\hat{\varepsilon}_{V,1}(2)$	$\hat{K}_1$
$\hat{\varepsilon}_{V,2}(1)$	$\hat{\varepsilon}_{V,2}(2)$	$\hat{K}_2$

instead of sort, sum the column of above, and take avarage

$$\frac{1}{Q}\sum_{q=1}^{Q} \min_{K=1\text{-}n} \frac{1}{Q}\sum_{q=1}^{Q} \| \hat{Y}^q_V - \hat{Y}^q_{V,K} \|^2 =$$

$$\frac{1}{Q}\hat{\varepsilon}_{V,Q}(K)$$

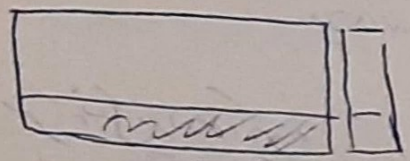
K might be very different.

How choose Q , q

repeat to stablize it

last ~~ap~~ caple of variation

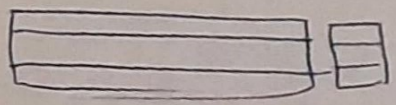
First: sequential version



Q fix

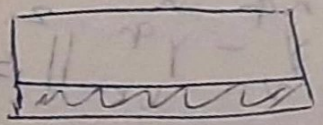
fix Π

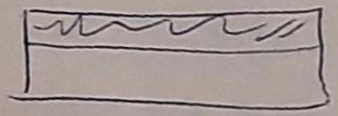
split را از فرکی سایتی کنی

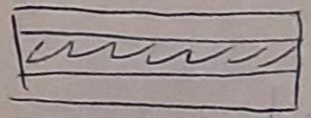


Q = 3

Q - FolD







n = 12 Q = 3 → training = 8 val = 4

Large or small Cardinality Cardinality

Q = 12 # train = 11 # val = 1

if you choose second, larger avarage.

Spliting is good for afew data.

leave-one-out ~~loo~~

Cross validation.

How choose algorithm:

1 split at random

Q split random

Q split sequentialy

may be the leave-One out.

1.5 oct

$(X_i, Y_i)_{i=1}^{n} \rightsquigarrow \hat{f}_K(x) = \sum_{i \in I_x} \frac{Y_i}{K}$ ✓

function

$\hat{X}, x, K \longrightarrow I_{x,K} \subset \{1 - n\}$

for $i = 1 \dots n$

$d_i := \| x - X_i \|$ $(d = d_1 \dots d_n)$

$(\tilde{I}, \tilde{d}) = \text{sort}(d \dots d_n)$

$\tilde{I}$: index of closes — get sorted number

$I_x = \tilde{I}(1-K)$

$k_{LN}(\hat{X}, \hat{Y}, x, K) = Y$

How do u choose K:

split $(\hat{X}, \hat{Y})$ — split, train: جدا کردن، آموزش دادن

$(\hat{X}, \hat{Y})$ train — $(\hat{X}_v, \hat{Y}_v)$ validation: اعتبارسنجی

Hold cross validation

$\hat{K} = \text{HO\_CV}(\hat{X}, \hat{Y})$

~~split $(\hat{X}, \hat{Y})$~~

$(\hat{X}_T, \hat{Y}_T), (\hat{X}_v, \hat{Y}_v) = \text{SPLIT}(\hat{X}, \hat{Y})$ ✓

for $K = 1 \dots n_{max}$ — for: حلقه

max n

① get

~~KNN~~

$KNN(\hat{X}_T, \hat{Y}_T, \hat{X}_v, K) = \hat{Y}_{v,K}$

$\epsilon_K = \| \hat{Y}_{v,K} - \hat{X}_v \|^2$

$(\tilde{I}(1), \epsilon(1)) = \text{Sort}(\epsilon)$ ← the first one

$\hat{K} = \tilde{I}(1)$

value sorted first is important

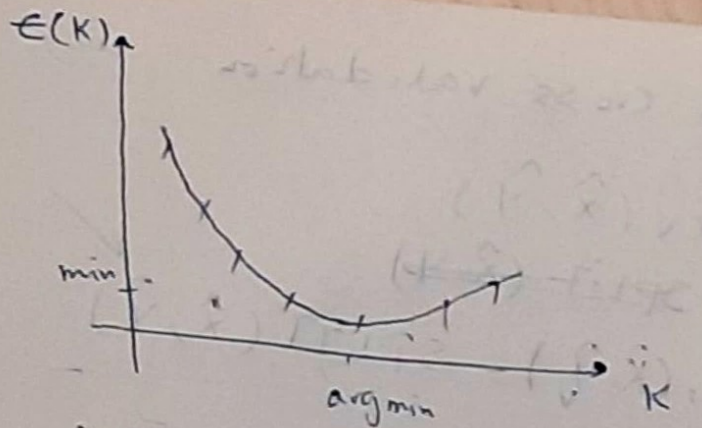
for $K = 1 \dots n$

$K\text{-}NN(\hat{X}_T, \hat{Y}_T, \hat{X}_v, K) = \hat{Y}_{v,K}$

$\epsilon(K) = \| \hat{Y}_{v,K} - \hat{X}_v \|^2 / N$

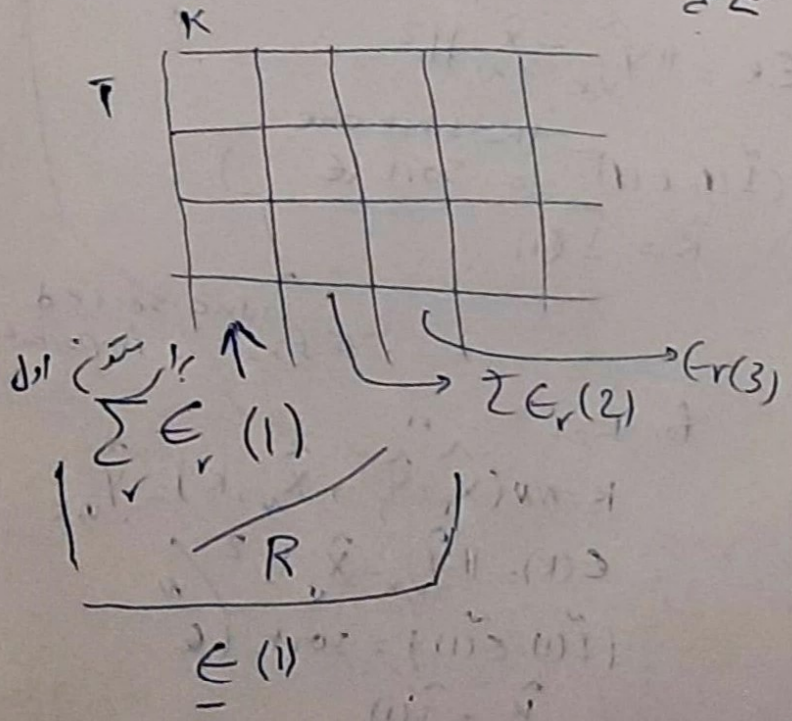
$(\tilde{I}(1)\ \tilde{\epsilon}(1)) = \text{sort}(\epsilon)$

$\hat{K} = \tilde{I}(1)$



← تکرار برای هر round و هر fold

$$\epsilon_r(k) = \| \hat{Y}_{v,k} - \hat{X}_v \|^2 / n$$ → ذخیره



not big dataset, so you must repeat after time

=

split.

10^6 داده → $\frac{1}{3}$ → 999999 + 1 → k_1
999998 + 2 → k_2

best of prediction

$$S_n = (X_i, Y_i)_{i=1}^n \sim P^n$$ ✓

$$\min_f \mathcal{E}(f) = \mathbb{E}_{(X,Y)\sim P} \left[(Y - f(X))^2 \right] =$$

$f: X \times Y$

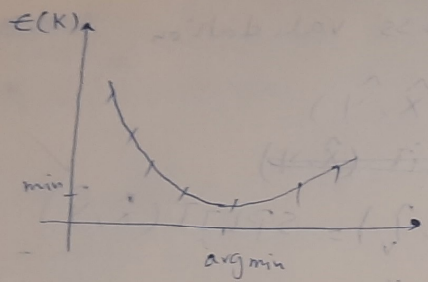
$$\int (Y - f(X))^2 \, p(x,y) \, dx\, dy$$

best of best

Can't Compute

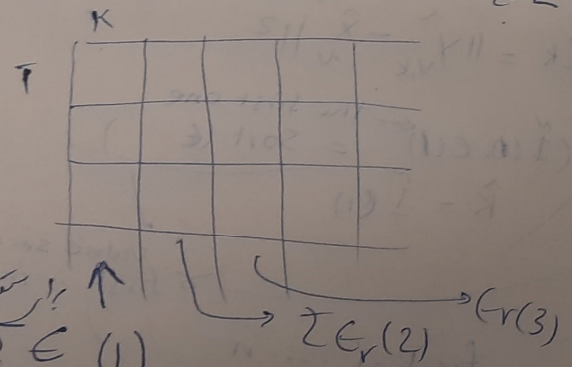
$$Y_i = f_*(X_i) + \delta_i \quad X_i \sim P_X \quad \delta_i \sim N(0, \sigma^2)$$

$$Y_i \sim N(f_*(X_i), \sigma^2)$$



← تکرار round برای هر for

$\epsilon_r(K) = \| \hat{Y}_{v,K} - \hat{X}_v \|^2 / n$ → برای



برای تکرار اول

$\sum_r \epsilon_r(1)$

R_r

$\epsilon(1)$

not big dataset, so you must repeat after time

split.

10^6 → 999999 + 1 → k_1

999998 + 2 → k_2

best of prediction

$S_n = (X_i, Y_i)_{i=1}^n \sim P^n$ ✓

$\min_{f \; X \times Y} \mathcal{E}(f) = \mathbb{E}_{(X,Y)\sim P} \left[(Y - f(x))^2 \right]$

$\int (Y - f(x))^2 \, p(x,y) \, dx\, dy$

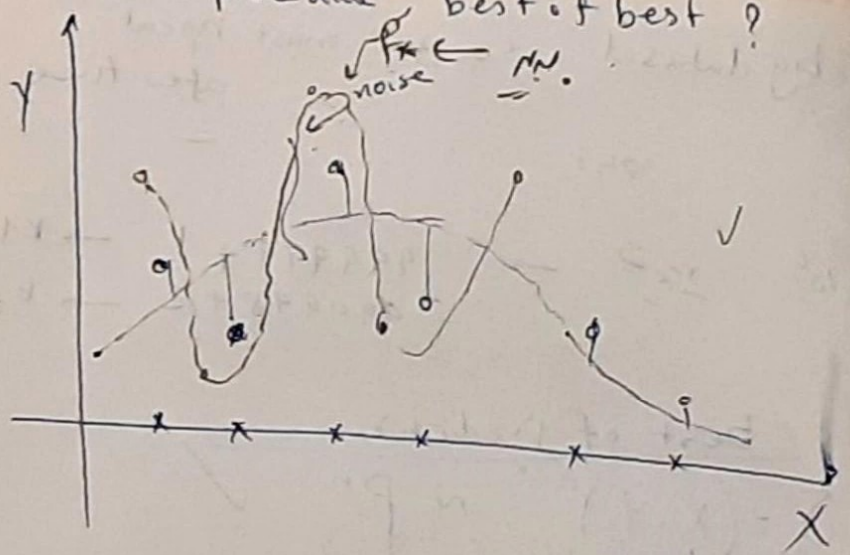
best of best

Can't Compute

$Y_i = f_*(X_i) + \delta_i \quad X_i \sim P_X \quad \delta_i \sim N(0, \sigma^2)$

$Y_i \sim N(f_*(X_i), \sigma^2)$

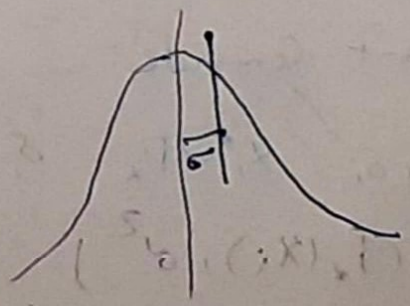
we have procedure, best of best ?



best : good prediction;

$$K = \arg\min_{K=1\dots n} \mathcal{E}(\hat{f}_K)$$

Can't Compute because we don't have infint data.



$$Y_i \sim N(f_x(X_i), \sigma^2)$$ IV

↑ noise level

که تابع رو بتونیم هموار انتظار خاصیتی خوشحالی

فرض بر همینه که noise = ۰ باشه

که الگوریتم رو بدیم ؛

$$f_K(x) = \frac{\sum_{i \in I_{x,K}} f_x(X_i)}{K} \sim f_x(X)$$

Compute X

define ✓

gentle function

can get very big change

$$\frac{L_x K}{n}$$

$$\left| f_K(x) = \frac{\sum_{i \in I_{x,K}} f_x(X_i)}{K} - f_x(x) \right|^2 \simeq \frac{L_x K}{n}$$

maximum change of function

هرچی زیاد و زیاد بشه ↑ خوبه مابه ازای بیشتری avarge

$$\frac{|f_x(x) - f_x(x')|}{|x - x'|} \leq L_x \quad \text{(I)}$$

I want to compare my algorithm with noise or not

$$\hat{f}_k(x) = \sum_{i \in I_{x,k}} y_i / k \quad \rightarrow \text{with noise according to (IV)}$$

$$f_k(x) = \sum_{i \in I_{x,k}} f_x(x_i) / k \quad \text{without noise}$$

$$\left|\hat{f}_k(x) - f_k(x)\right|^2 \lesssim \; ?$$

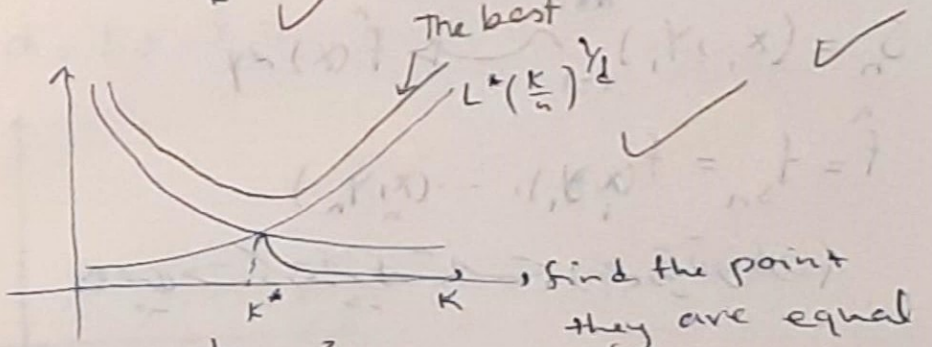
$$= \sum_{i \in I_{x,k}} \frac{\delta_i}{k} \rightarrow \text{variance}$$

$$\mathbb{E}\left|\hat{f}_k(x) - f_k(x)\right|^2 = \frac{1}{k^2}\mathbb{E}\left|\sum_{i \in I_{x,c}} \delta_i\right|^2 = \frac{\sigma^2}{k}$$

$$\sum_{i \in I_{x,k}} \qquad k\sigma^2$$

$$\left|\hat{f}_k(x) - f_k(x) \pm f_k(x)\right|^2 \lesssim L^* \left(\frac{k}{n}\right)^{\frac{1}{d}} + \frac{\sigma^2}{k}$$

good guy



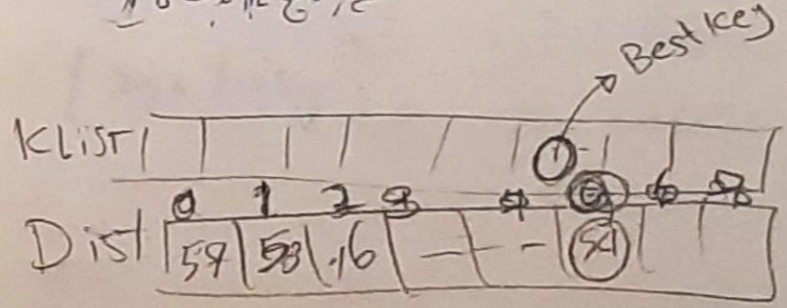
, find the point they are equal

$$L^* \left(\frac{k}{n}\right)^{\frac{1}{d}} = \frac{\sigma^2}{k}$$

$$k_d = \left(\frac{\sigma^2 n^{1/d}}{L^*}\right)^{\frac{d}{d+1}}$$

$\hat{k}$ is good way to find good guy

when function change?



$(x_i, y_i)_{i=1}^{m} \rightsquigarrow \hat{f}(x) \sim y$

↑ Important

$S_n = (x_i, y_i)_{i=1}^{n} \rightsquigarrow \hat{f}(x) \sim y$

$\hat{f} = f_{S_n} = f_{(x_1,y_1),\dots(x_n,y_n)}$

(این صورت که مجموعه داده است را می‌گیرد)

$S_n = (x_i, y_i)_{i=1}^{n} \sim P^n \rightarrow \hat{f}(x) \sim y \; , \; X \in \mathbb{R}^d, y \in \mathbb{R}$

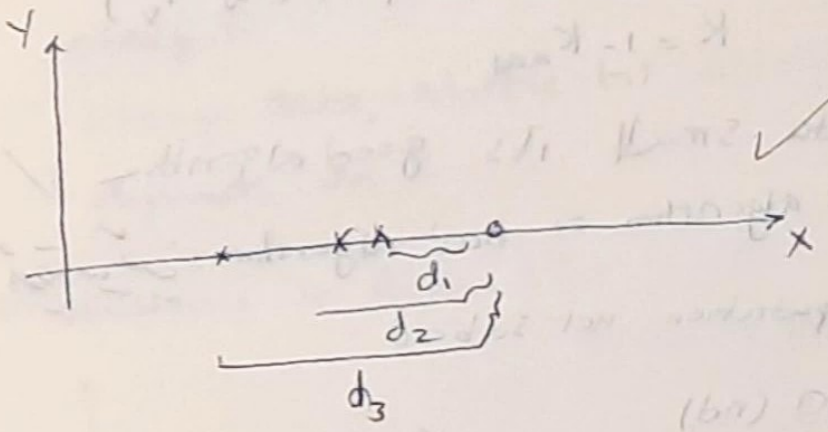
solve

$\min_{f: X \to \mathbb{R}} L(f) \qquad L(f) = E_{(x,y)\sim P}\left[\ell(f(x), y)\right]$

given $S_n \sim P^n$

$$\hat{f}_K(x) = \frac{1}{K} \sum_{i \in I_x} y_i$$

$d_i = \|x - x_i\| \; , \; d = (d_1 \dots d_n)$



$(\tilde{I}, \tilde{d}) = \text{Sort}$

$I_x = \tilde{I}(1 \dots K)$

if the laybel of y devide by K :

$$[\text{Sign } f(x) =]_{-1}^{+1} = \begin{cases} 1 & f(x) > 0 \\ -1 & f(x) \leq 0 \end{cases}$$

finding K:

$K = Ho\_CV(\hat{x}, \hat{y})$ ✓

$split(\hat{x}, \hat{y}) \rightsquigarrow (\hat{x}_T, \hat{y}_T), (\hat{x}_V, \hat{y}_V)$

$K = 1 - K_{max}$

data small its good algorithm ✓

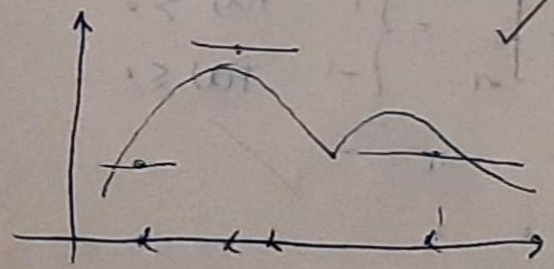
good algorithm or bad algorithm; بستگی

Computation not so bad.

O:(nd) ✓

big data,

nearest neighbour is flexible if you have enough data. ✓



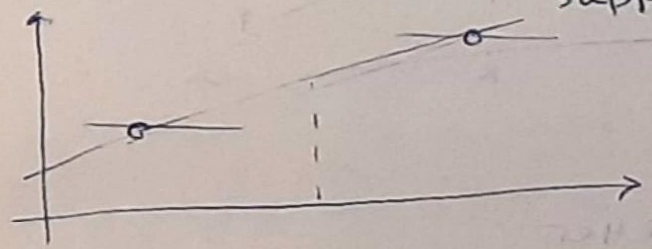
KNN can approximate many functions provided enough data.

nervous! : (KNN is very nervous!

change K ✓

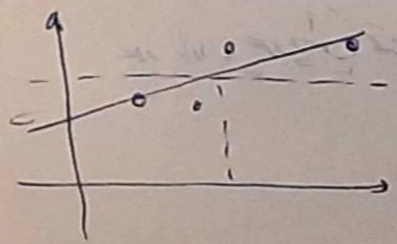
change data, a little bit always depends on K

sensetive to data! (for small K)



the only way to start to best is

بستگی دارد KNN بهترین راه برای شروع



? small data ↪ line

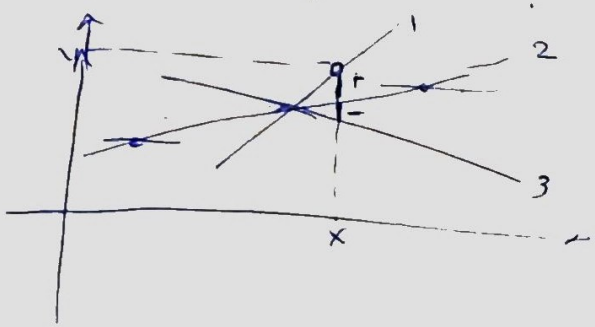
big data ↪ K-NN

یعنی کران انتگرال ضبط حرف است.

to find line : 2point

finding line is global _ not good
not bad

if I assume linear model this is global, as present ...?



2 is better

$Wx+b \rightarrow$ خط

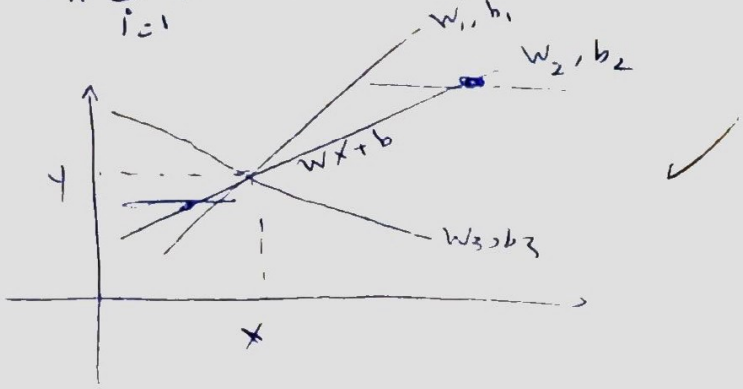
Wx_i+b-y_i

عدد بالا، عددی است که میزان فاصله بیشتر از خط را نشان می‌دهد

$$\sum_{i=1}^{n} |Wx_i+b-y_i|^2$$

یا میانگین فاصله را حساب می‌کنیم

$$\frac{1}{n}\sum_{i=1}^{n} |Wx_i+b-y_i|^2$$



I judge line if close to my data or not

$$\frac{1}{n}\sum_{i=1}^{n} (Wx_i+b-y_i)^2$$

How to you find good W

delete b:

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

lets write function and take derivative.

function of W ✓

Lost of data

$F(W)$, instructive name is $L(W)$

$\hat{L}(W)$ is agreed name for loss fun

of data $\Rightarrow \frac{1}{n}\sum_{i=1}^{n}(wx_i - y_i)^2 = \hat{L}(w)$

If: $\frac{d\hat{L}(w)}{dw} = \frac{1}{n}\sum_{i=1}^{n} 2x_i(wx_i - y_i) = 0 \Rightarrow$

$\frac{2}{n}\sum_{i=1}^{n} x_i(wx_i - y_i) = 0 \Rightarrow (\sum_{i=1}^{n} x_i^2)w = \sum_{i=1}^{n} x_i y_i$

$\hat{w} = LS(\hat{x}, \hat{y})$ ؟!

آیا الگوریتم در اسپانیا درست خواهد بود: به خوبی

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

scaler product, Turn W to W^T to make scaller

توابع برداری،

derivative of many dimention its vector first of all built vector, freakency gradient

I:

$\nabla\hat{L}(w) = \begin{cases} \frac{\partial}{\partial w^1}\hat{L}(w) = \frac{2}{n}\sum_{i=1}^{n} 2(w^T x_i - y_i)^2 \\ \frac{\partial^d}{\partial w^d}\hat{L}(w) = \frac{1}{n}\sum_{i=1}^{n} 2(w^T x_i - y_i) \end{cases}$

$w^T x = \sum_{i=1}^{d} w^i x^i$

$\Rightarrow$

$$\nabla \hat{L}(w) = \begin{cases} \frac{\partial}{\partial w^1} \hat{L}(w) = \frac{1}{n} \sum 2 x_i^1 (w^T x_i - y_i)^2 \\ \frac{\partial}{\partial w^d} \hat{L}(w) = \frac{1}{n} \sum_{i=1}^{n} 2 x^d (w^T x_i - y_i) \end{cases}$$

$$\nabla w^T x = \begin{cases} x \\ x^d \end{cases} = x$$

$$\nabla \hat{L}(w) = \frac{2}{n} \sum_{i=1}^{n} x_i (w^T x_i - y_i)$$

$$F(x) = F(x) = w^T x = \sum_{j=1}^{n} w^j x^j \quad \leftarrow \text{Transpose}$$

$$\nabla F(w) = \left(\frac{\partial F(w)}{\partial w^1}, \ldots, \frac{\partial F(w)}{\partial w^d} \right)$$

$$\Rightarrow \nabla w^T x = x$$

which means: ✓

$$\nabla \hat{L}(w) = \begin{cases} \frac{\partial}{\partial w^1} \hat{L}(w) = \frac{1}{n} \sum_{i=1}^{n} 2 x_i^1 (w^T x_i - y_i)^2 \\ \frac{\partial}{\partial w^d} \hat{L}(w) = \frac{1}{n} \sum_{i=1}^{n} 2 x^1 (w^T x_i - y_i) \end{cases}$$

$$\frac{2}{n} \sum_{i=1}^{n} x_i (w^T x_i - y_i) = 0$$

$$\sum_{i=1}^{n} x_i (w^T x_i) = \sum_{i=1}^{n} x_i x_i^T w \quad *$$

and $z = \text{cov}(x_1 \ldots x_m, w)$ ✓

$$(\ldots) \sum_{i=1}^{n} x_i x_i w = \sum_{i=1}^{n} x_i y_i$$

Compute vector 2ways: (sudocode has 2 ways)

$*$: $\underbrace{\underbrace{x_i^T}_{1 \times d} \underbrace{x_i}_{d \times 1}}_{1}$, $\underbrace{\underbrace{x_i}_{d \times 1} \underbrace{x_i^T}_{1 \times d}}_{d \times d}$

$\Rightarrow$

for $i = 1 \dots n$

$$\hat{C}_i = \hat{C}_{i-1} + X_{i-1} X_{i-1}^T$$

$$S_i = S_{i-1} + X_{i-1} (X_{i-1}^T w)$$

Code

Linearity of these operations

$$\Rightarrow (\dots)\left(\sum_{i=1}^{n} X_i X_i^T\right) w = \sum_{i=1}^{n} X_i y_i = h$$

$$\hat{C} \hat{w} = \hat{h}$$

taking gradient, of linear function -

$$\hat{C} w = \hat{y}$$

$d \times d \quad d \times 1 \quad d \times 1$

Linear system.

" $\hat{w} = \hat{C}^{-1} \hat{y}$ "

it should be invertable

$\Rightarrow$ You have code.

task { $\hat{w} = LS(\hat{x}, \hat{y})$; compute $\hat{C}, \hat{y}$; $\hat{w} = \hat{C}^{-1} \hat{y}$

2 minutes

(Probability ... $L(f)$ is too ...)

**

$$\hat{L}(f) = \frac{1}{n} \sum_{i=1}^{n} \ell(f(x_i), y_i)$$

Solve

$$\min_{f: X \to \mathbb{R}} L(f) \qquad L(f) = \mathbb{E}_{(x,y)\sim P} [\ell(f(x), y)]$$

EMPIRICAL RISK MINIMIZATION

$$\min_{f_w(x) = w^T x} \hat{L}(f_w) \qquad \hat{L}(f) = \frac{1}{n} \sum_{i=1}^{n} \ell(f(x_i), y_i)$$

$\ell(a,b) = (a-b)^2$ Least - SQUARES

✓

LEAST SQUARES - ERM

$$\begin{cases} \min L(f) \\ f: X \to Y \\ \text{given } (x_i, y_i)^n \sim P^n \end{cases} \qquad L(f) = \mathbb{E}_{(x,y)\sim P} [\ell(f(x), y)]$$

$$\begin{cases} \min \hat{L}(f_w) \\ f_{(w)}(x) = w^T x \\ w \in \mathbb{R}^d \end{cases} \qquad \hat{L}(f) = \frac{1}{n} \sum_{i=1}^{n} \ell(f(x_i), y_i)$$

$\ell(a,b) = \ell(a,b)^2$

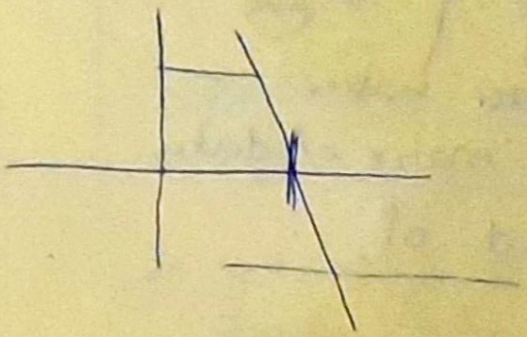
$\ell(f(x), y) = (f(x) - y)^2$

~~$\ell(a,b)$~~

$y \in \{\pm 1\} = (yf(x) - 1)^2$

$f(x) = w^T x \qquad f: X \to \{\pm 1\}$

even if data look like.



$\text{sign}(f(x)) = 1$

$\text{sign}(f(x)) = -1$

① $l(f(x), y) = (f(x) - y)^2$

② $y \in \{\pm 1\} = (y f(x) - 1)^2$

$(yf(x) - y)^2 = (1 - yf(x))^2$

$= (1 - yf(x))^2$

---

* $\nabla \hat{L}(w) = 0 \Rightarrow \hat{C} w = \hat{h}$

$\hat{C} = \frac{1}{n} \sum_{i=1}^{n} x_i x_i^T = \frac{1}{n} \hat{X}^T \hat{X}$ — $\hat{X}$: contain all input as rows & column each simple coordinate across; dim: d × n

$\hat{h} = \frac{1}{n} \sum_{i=1}^{n} x_i y_i = \frac{\hat{X}^T \hat{Y}}{n}$

x_i: vector, y_i: number

Covariance matrix of data

second moment of ...

$\frac{1}{n} \sum_{i=1}^{n} x_i = 0$

---

* I'm Looking for w

$\hat{w} = \hat{C}^{-1} \hat{h}$

$\nabla \hat{L}(w) = 0 \Rightarrow \hat{C} w = \hat{h}$

$\frac{1}{n} \| \hat{X} w - \hat{Y} \|^2$

$\hat{X} w = \hat{Y}$

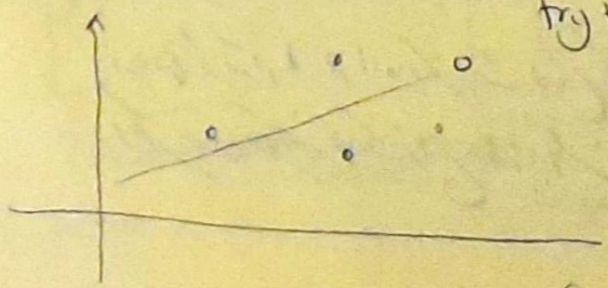
if w satisfy of this exist →

$\hat{L}(w) = 0$

$\hat{X} w = \hat{Y} \iff x_i^T w = y_1$ (**)

$\vdots \quad \vdots$

$y_n^T w = y_n$



try to find line through node

do you think you find w that satisfy (**) : generally no !

according to the graph:

$n=5$, $d=1$

$n > d$

number of w is less than the number of point we called

under parameterized

$n > d$ over determined

$n < d$ under "low" "

Interpolation

$\frac{d-n}{\text{Solution}}$ we have

$n < d$ ~~Can~~ Can plot ~~interpol~~

اون کنترل از اسارکریپت لیبل شده

دیگر سی توی زیرکنترل نظارت نیست

one thing you can do

$\hat{C}$ is defined this way:

$$\hat{C} = \frac{1}{n} \sum_{i=1}^{n} X_i X_i^T = \frac{1}{n} \hat{X}^T \hat{X}$$

if matix is full rank its invertible

can invert it happy.

2 different Problem not two different matrix

~~what is the~~

$n > d$

X' $\hat{X}$ $= \hat{C}$ dxd ①

$n > d$ ~~need~~ ↘ squre is dependent

$n \times d$

$n < d$

X^T $\hat{X}$ $= \hat{C}$ dxd ②

I dont have square number here ↑ huge matrix

↘ is squre is independent

What is Rank of * , **

for * : Rand is d

for ** : Rand is n

how to fix this:

Change the Algorithm:

$$\min_{w \in \mathbb{R}^d} \hat{L}(w) + \lambda \|w\|^2$$

λ: Spoil

$\|w\|^2$: square norm

$\hat{w}$ کی، w کی
کرنے کو ہم کیا کرنا چاہتے ہیں

$F(\theta)$

$$\min F(\theta) = F(\theta_*)$$

in our case

$$\min_w \hat{L}(w) = \hat{L}(\hat{w})$$

پیشگوئی شدہ
$\hat{w}$ شدہ پیشگوئی

what is going on here?

the idea is minimize error

$$\min_w \hat{L}_\lambda(w) = \lambda \|w\|^2, \quad \hat{L}(w) = \frac{1}{n}\sum_{i=1}^{n}(Y_i - w^T X_i)^2$$

$$\hat{L}_\lambda(w)$$

$$\nabla \hat{L}_\lambda(w) = 0 \Rightarrow \hat{C}w = \hat{h}$$

$$\nabla \hat{L}(w) = +2\hat{X}^T(\hat{X}w - \hat{Y}) = +2(\hat{C}w - \hat{h}) \quad (I)$$

$$\nabla \|w\|^2 = 2w$$

λ is constant

remember that is a norm

$$\|w\|^2 = \sum_{j=1}^{d}(w^j)^2 \Rightarrow$$

from (I)

$$\nabla \hat{L}_\lambda(w) = 2(\hat{C}w - \hat{h}) + \lambda 2w$$

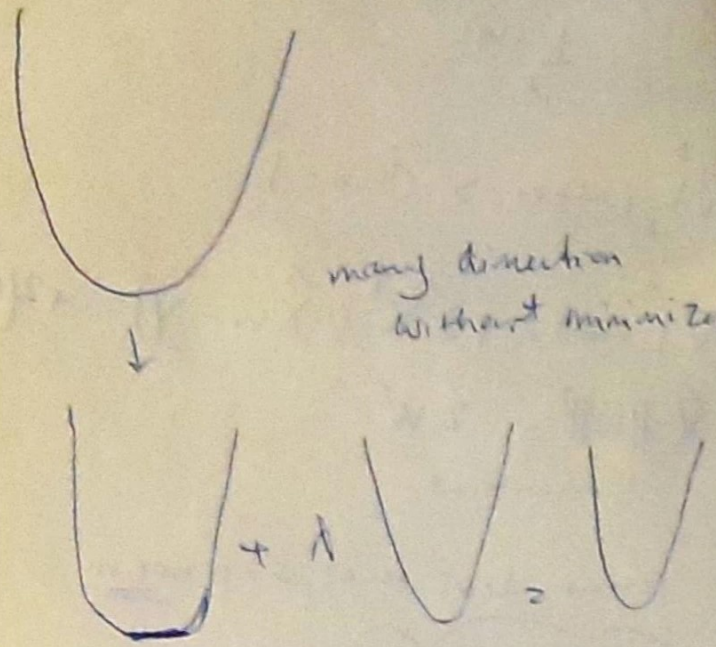
$$= 2(\hat{C} + \lambda I)w - 2\hat{h} = 0$$

$$(\hat{C} + \lambda I)w = \hat{h}$$

$$\hat{w}_\lambda = (\hat{C} + \lambda I)^{-1}\hat{h}$$

if you want to draw in one dimension

what kind of function is this:



first observation

larger λ

smaller λ

Second observation

$$\hat{w} = (\hat{C} + \lambda I)^{-1} \hat{h} \quad II$$

$$\hat{C} \to (\hat{C} + \lambda I)$$

$$\hat{C} = \begin{pmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_d \end{pmatrix}, \quad \hat{C} \mapsto (\hat{C})^{-1}$$

$$\hookrightarrow \begin{pmatrix} \frac{1}{\sigma_1} & & \\ & \frac{1}{\sigma_2} & \\ & & \frac{1}{\sigma_d} \end{pmatrix}$$

if rank < d

λ machover of value,

if $\hat{C} \to (\hat{C} + \lambda I)^{-1}$

$$\begin{pmatrix} \frac{1}{\sigma_1+\lambda} & & \\ & \frac{1}{\sigma_2+\lambda} & \\ & & \frac{1}{\sigma_n+\lambda} \end{pmatrix}$$

$\sigma \gg \lambda \quad \frac{1}{\sigma+\lambda} \sim \frac{1}{\sigma}$

$\sigma \ll \lambda \quad \frac{1}{\sigma+\lambda} \sim \frac{1}{\lambda}$

Small value I care

In special matrix, you can see what we did.

Uniqness & invertibility ?

its better λ is big or small

2 question: how to find λ

$\hat{C}$ is not diagonal

never complex

$$\begin{pmatrix} \sigma_1 & & & \\ & \sigma_2 & & \\ & & \sigma_3 & \\ & & & \sigma_d \end{pmatrix} V^T = V \begin{pmatrix} \frac{1}{\sigma_1+\lambda} & & & 0 \\ & \frac{1}{\sigma_2+\lambda} & & \\ & & \frac{1}{\sigma_3+\lambda} & \\ 0 & & & \frac{1}{\sigma_d+\lambda} \end{pmatrix}$$

D

how to solve problem in practice

- solve linear system II

built C

built h

$\hat{w}_\lambda = RLS(\hat{X}, \hat{Y}, \lambda)$

$\hat{C} = \hat{X}^T X$

$\hat{h} = \frac{\hat{X}^T \hat{Y}}{m}$

$[D, V] = eig(\hat{C})$

↓ procedure in numpy which return (D)

$B = V(D+\lambda I)^{-1} V^T$

$\hat{w}_\lambda = B\hat{h}$

instead of this

$w = \hat{h}(\hat{C} + \lambda I)$

↑ not inverse

Logestic Regression & SVM

ERM Empirial Risk Minimization

$$\left[\begin{array}{l} \min \mathcal{E}(f) \quad \mathcal{E}(f) = \mathbb{E}_{(x,y)\sim P}\left[\ell(f(x),y)\right] \\ f: X \to \mathbb{R} \\ \text{given } (x_i,y_i)_{i=1}^{n} \sim P^n \end{array}\right]$$

$$f_w(x) = w^T x \qquad n < d$$

$$\hat{\mathcal{E}}(f) = \frac{1}{n}\sum_{i=1}^{n} \ell(f(x_i), y_i)$$

$$\min_{w\in\mathbb{R}^d} \underbrace{\frac{1}{n}\sum_{i=1}^{n} \ell(w^T x_i, y_i)}_{\hat{\mathcal{E}}(w)}$$

$$f(x) = \underset{w}{w^T x} + b$$

$$(w, b) \leftarrow w^T x \sim (x, 1)$$

$$\ell(w^T x, y) = (w^T x - y)^2$$

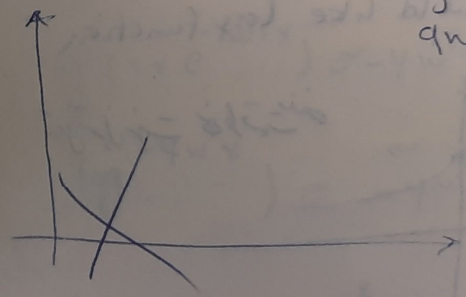
$$\min_{w\in\mathbb{R}^d} \underbrace{\frac{1}{n}\sum_{i=1}^{n} \ell(w^T x_i, y_i) + \lambda \|w\|^2}_{\hat{\mathcal{E}}(w)}$$

دوروش دیگر وجود دارد که نمیدانم چیست، فقط میدانم که

logistic / cross entropy loss $= \log(1 + e^{-y w^T x})$

How come up with sth like this:

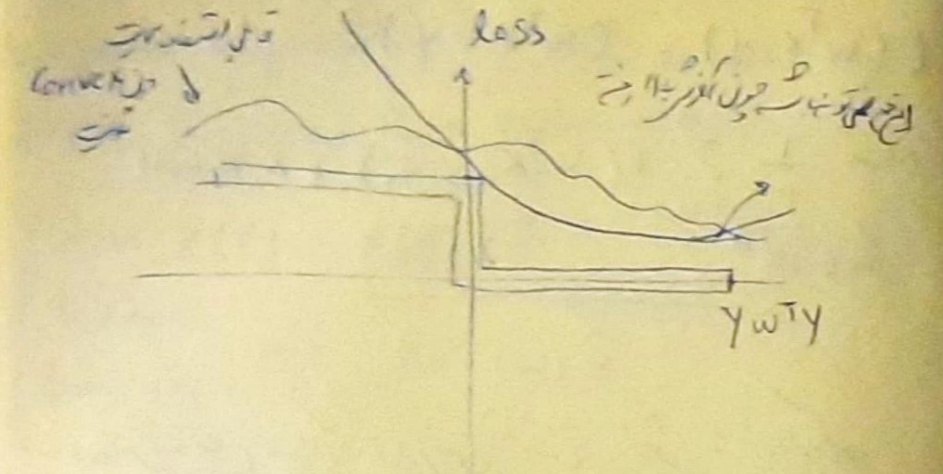
figure out how to plot quantity to check is $-y w^T x$



- we want to plot logistic / cross entropy

to plot we saw log — —

think how to draw according $-y w^T x$

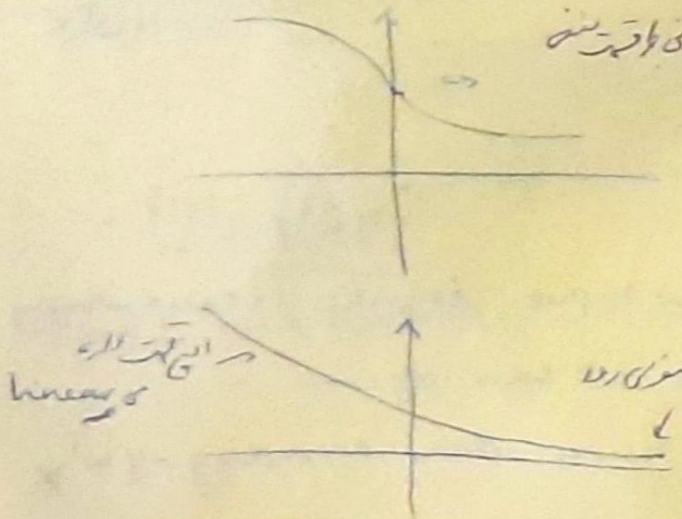


$D\, y w^T x \leq 0$

2 problem: Smoothness, Convexity

flat ضعیف

Second how would like loss function

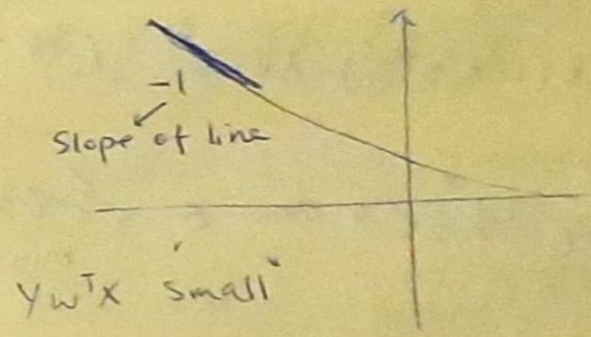


when $y w^T x$ is very small

$$\log\left(1 + \frac{1}{e^{y w^T x}}\right)$$

$e^{y w^T x} \to \infty$

close to $+\infty$



$\log(1 + e^{-y w^T x}) \simeq -y w^T x$

$\log^{-y w^T x}) \simeq -y w^T x$ (like a line)

show

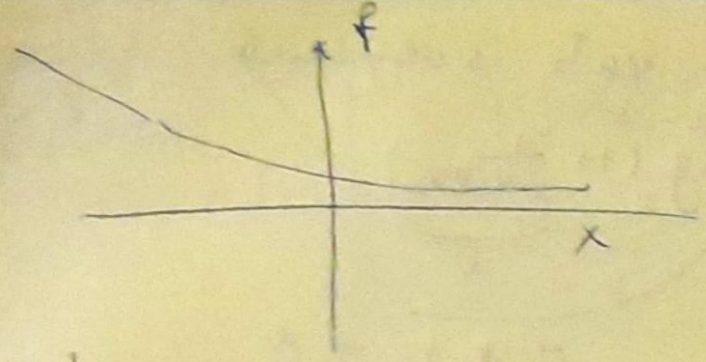
$f(x) = \log(1 + e^{-x})$

$x \to -\infty$

$\log(1 + e^{-x}) \simeq \log(e^{-x})$

$x \to \infty \quad x \quad -x$

$\log(1 + e^{-x}) \simeq \log 1 \simeq 0$



$\log(1+e^{-x})$

$$\min_{w \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^{n} \ell(w^T x_i, y_i) + \lambda \|w\|^2$$

$$\nabla_w \hat{\varepsilon}(w) =$$

$$\nabla_w \|w\|^2 =$$

$$\nabla_w \hat{\varepsilon}(w) = \nabla_w \frac{1}{n} \sum_{i=1}^{n} \log\left(1+e^{-y_i w^T x_i}\right)$$

$$= \frac{1}{n} \sum_{i=1}^{n} \nabla_w \log\left(1+e^{-y_i w^T x_i}\right)$$

$$= \frac{1}{n} \sum_{i=1}^{n} \frac{1}{1+e^{-y_i w^T x_i}} \cdot \left(-e^{-x}\right)$$

$$= \frac{1}{n} \sum_{i=1}^{n} \frac{e^{-y_i w^T x_i}}{1+e^{-y_i w^T x_i}} \cdot (y_i x_i)$$

$$= \frac{1}{n} \sum_{i=1}^{n} -y_i x_i \Big/ 1+e^{y_i w^T x_i}$$

$\Rightarrow$ we descovered the gradian

$$\nabla_w \hat{\varepsilon}(w) = \frac{1}{n} \sum_{i=1}^{n} \frac{-y_i x_i}{1+e^{y_i w^T x_i}}$$

$$\nabla_w \hat{\varepsilon}(w) + \lambda \|w\|^2 = \frac{1}{n} \sum_{i=1}^{n} \frac{-y_i x_i}{1+e^{y_i w^T x_i}} + 2\lambda w = 0$$

question is what do we do?

Find w,?

Linear system, a system of linear equation

$\frac{1}{e}$... not linear, non linear in w

No closed From Solution!

optimal condition

I want squar lot to get linear system
linear system is bucket

Gradient Methods

(first order method)
use
first derivative not second derivitive,
so call it first order:

$w_{t+1} = w_t - \gamma_t \nabla_w \hat{\varepsilon}(w_t)$ $\quad \gamma_t$ = Constant
$w_0 = 0$

Logistic_train $(\hat{X}, \hat{Y}, \lambda)$

~~compute~~
$w_0 = 0$, $\gamma_t = (?)$
for $t = 1 \text{ — } T_{MAX}$
$g(t) = \nabla \hat{\varepsilon}_\lambda(w_t)$;
$w_{t+1} = w_t - \gamma_t g(t)$

We want w to find the main algorith
minimize.

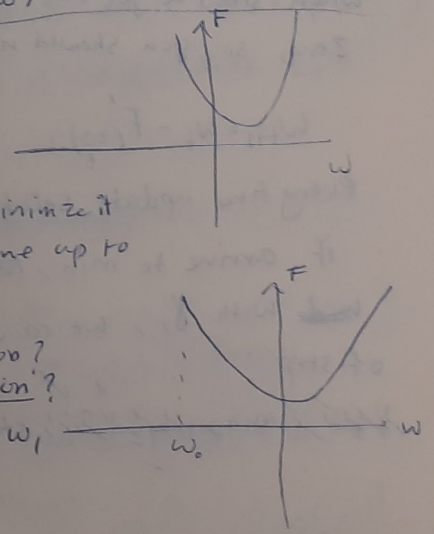
$$\nabla_w \left[\hat{\varepsilon}(w) + \lambda \|w\|^2 \right] = \frac{1}{n} \sum_{i=1}^{n} \frac{-y_i x_i}{1 + e^{-y w^T x_i}} + 2\lambda w = 0$$

$\hat{\varepsilon}_\lambda(w)$ $\qquad$ $\nabla \hat{\varepsilon}_\lambda(w)$

$$\min_{w \in \mathbb{R}^d} \underbrace{\frac{1}{n} \sum_{i=1}^{n} \ell(w^T x_i, y_i) + \lambda \|w\|^2}_{\hat{\varepsilon}(w)}$$

$F(w)$

$F: \mathbb{R}^d \to \mathbb{R}$
$d = 1$
there is function
& I want to minimize it
we want to come up to
algorithm.
invent a irritation?
<u>niteration</u>?
go from w_0 to w_1
$w_0 \to w_1$



take a point near w_0
$\min F(w_i) \leq F(w_0)$
check if $F(w_1) \leq$

می‌خواهم از w_0 به w_1 بروم چک کنیم که [illegible] مناسب [illegible]

F را که مشتق را [illegible] مقدارش منفی باشد $F(w)$ کم شود [illegible]

[illegible] است

یک راه هم هست که مشتق بگیریم و در جهت خلاف آن حرکت کنیم

$$w_1 = w_0 - F'(w_0)$$

when start to get to min, gradian is Zero, so you should not be worry.

$$w_{t+1} = w_t - F'(w_t)$$

every time update solution, get steps

if arrive to min, how to control ~~wid~~ with γ, we can control size of step.

با مقدار گاما می‌توانیم step را ریزتر بکنیم ولی در min که [illegible]

[illegible] نشیم.

When function change,

first derivative change alot,

Second derivative is big you want step to be small

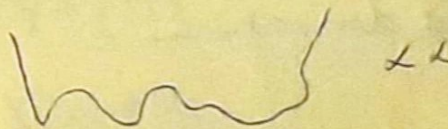
newton comes up to this equation:

$$w_{t+1} = w_t - \frac{F'(w_t)}{F''(w_t)}$$

~~[illegible]~~ γ is not constant hear.

local search is good γ good for ∇

با مقدار γ تغییرات را [illegible]



| Newton method ~~or~~ in $d=1$ | (NM)

if dimention is begger than 1,

NM $d>1$

~~$w_{t+1} = w_t \; \nabla F(w_t)$~~

~~Hessian~~ H

$$w_{t+1} = w_t - \underbrace{H(w_t)^{-1}}_{\text{inverse}} \nabla F(w_t)$$

inverse Hessian at w_t!

Hessian $H(w)$

ATM

in practice, to break it enough

Gradient descent

$w_{t+1} = w_t - \gamma_t \nabla F(w_t)$

(1) $\gamma_t \to 0$

(2) if you know maximum change in ~~maximum~~ second derivative

in (1) $\gamma_t \to 0$ Decaying step size

$\forall w, \underline{w}$

$\|\nabla F(w) - \nabla F(\underline{w})\| \leq c \|w - \underline{w}\|$

maximum change in 1st derivate

$\longleftrightarrow$ get second derivative

if you take square

$\gamma = \frac{1}{2c}$

$F(w_t) - \min_{w \in \mathbb{R}^d} F(w) \lesssim \begin{cases} \frac{1}{t} & \lambda = 0 \\ e^{-t} & \lambda > 0 \end{cases}$

λ large better ??

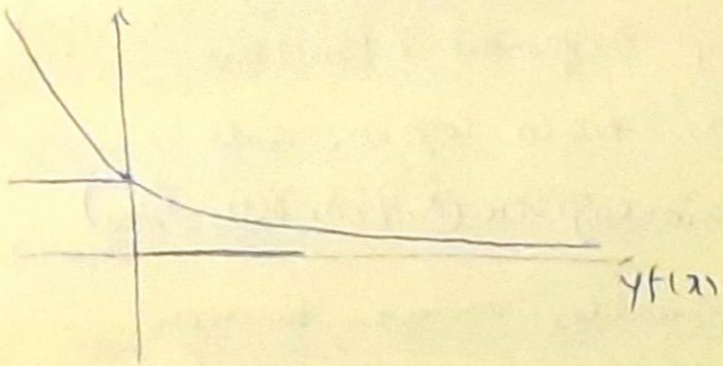
~~xxx~~

$\|\nabla \hat{\mathcal{E}}_\lambda(w) - \nabla \hat{\mathcal{E}}_x(\underline{w})\| \leq c \|w - \underline{w}\|$?

Logistic Regression ~ SVM (Support vector Machines)

$y \in \{\pm 1\}$

loss function $\ell(y \cdot f(x))$



ERM

$$\min_{w \in \mathbb{R}^d} \underbrace{\frac{\sum_{i=1}^n \ell(y_i, w^T x_i)}{n} + \lambda \|w\|^2}_{\mathcal{E}_\lambda(w)}$$

$\log(1 + e^{-yf(x)})$

$$\nabla \hat{\varepsilon}_\lambda(w) = \frac{1}{n} \sum_{i=1}^{n} \frac{y_i x_i}{1 + e^{y_i w^T x_i}} + 2\lambda w = \phi$$

~~$w_0 = 0$, $w_{t+1} = w_t$~~

$w_0 = 0$, $w_{t+1} = w_t - \gamma_t \nabla \hat{\varepsilon}_\lambda(w_t)$, T_{max} nd

have d number against d number

cost of this nd intime

time complexity

how many argument ɣ function

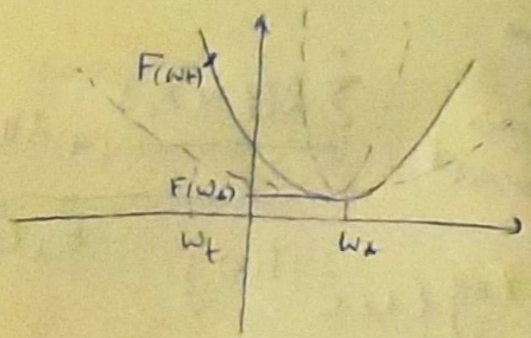
get w, train logistic code

$w = \text{Train-Logistic}(\hat{x}, \hat{y}, \lambda, \gamma(t), T_{max}, \tau)$

think generalize, minize function

$F: \mathbb{R}^d \to \mathbb{R}$

$\min_{w \in \mathbb{R}^d} F(w)$



w_t, compute Gradient:

$\nabla F(w_t) \leq \tau$

↑

below τ

$F(w_*) = \min_{w \in \mathbb{R}^d} F(w)$ *

$\begin{cases} F(w_t) - \min_{w \in \mathbb{R}^d} F(w) \leq z \\ \| w_t - w_* \| \leq z \end{cases}$ ** , *** *

$F(w_t) - F(w_{t+1}) \leq z$

You can not compute

$\| w_t - w_* \|$

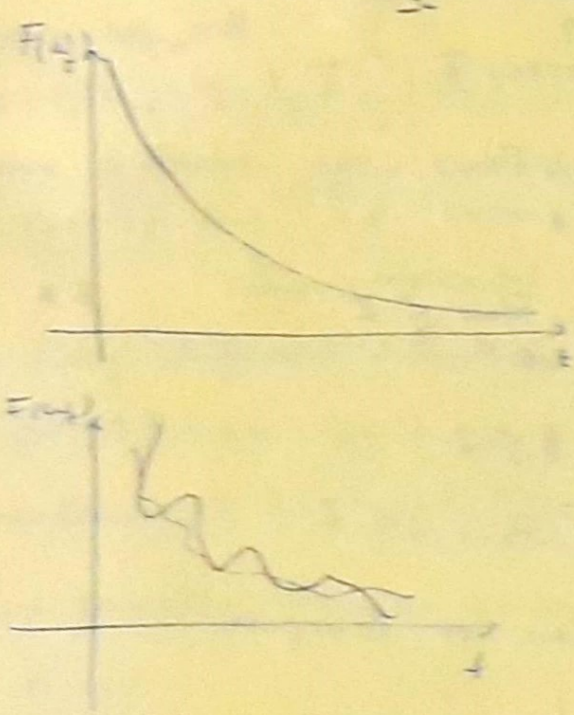
look at { : you start from w_t ɣ get

$F(w_t)$,

according to diagram,

ازشکل یک از موردی ... ازش آگوده استفاده کنیم آیات

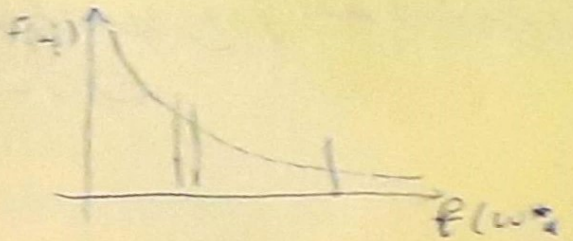
وتاآیات



So we can have

$w = \text{Train\_logistic}(\hat{X}, \hat{Y}, \lambda, \gamma(1), \tau, T_{max})$

threshold



Step size

$w_{t+1} = w_t - \gamma_t \nabla F(w_t)$

γ_t if $d=1 \Rightarrow \gamma_t = \frac{1}{F''(w_t)}$

$F''(w_t)$ because

how much derivative changed

$d>1$, $\gamma_t = H(w_t)^{-1}$ $O(d^3)$!!

- $nd \longrightarrow dq$ with every iteration

How do break it ?

smaller step ?

$\frac{|F'(w) - F'(v)|}{|w-v|} \leq C$

in any dimention

$\frac{|\nabla F(w) - \nabla F(v)|}{|w-v|} \leq c$

if Σ face this: $\|\nabla \hat{E}(w) - \nabla \hat{E}(v)\| \leq$

$\| \frac{1}{n} \sum_{i=1}^{n} \frac{-y_i x_i}{1+e^{y_i w^T x_i}} - \frac{1}{1+e^{y v^T x_i}} \| + 2\lambda \|w-v\|$

Suppose big data set 10,000,000 point

dimention ~~10000~~ 10^6 10^3 1000

$O(nd)$

how can we breack this.

$N_0 = 0 \quad W_{t+1} = W_t - \gamma_t \nabla \hat{\mathcal{E}}_\lambda(w_t)$

if have 1000000 point in 1000 dimantion

how much memory do you need?

when we are working in hurry

data set big, not time to have iteration

if you don't have memory how you would implement?

Split it , I can't compute ∇ ---

Compute this for a part of data

for b point

$$\nabla \hat{\mathcal{E}}_\lambda(w_t) = \frac{1}{b} \sum_{i=1}^{b} \frac{-y_i x_i}{1+e^{y_i w_t^T x_i}} + 2\lambda w_t \neq 0$$

$g(t)$

compute a solution, exactly

must comput 1000

$\times$ 1000

$\times$ 1000

do γ update solution.

say batch ل

$$\nabla \hat{\mathcal{E}}_\lambda(w_1) = \frac{1}{b} \sum_{i=1}^{b} \frac{-y_i x_i}{1+e^{y_1 w_1^T x_i}} + 2\lambda w_1 \neq 0$$

$w_1 \rightarrow$

for w_2 batch ل

$$\nabla \hat{\mathcal{E}}_\lambda(w_2) = \frac{1}{b} \sum_{i=b+1}^{2b} \frac{-y_i x_i}{1+e^{y_i w_2^T x_i}} + 2\lambda w_2 = 0$$

$\longrightarrow$ so for t (ل t) batch = t

$$\nabla \hat{\mathcal{E}}_\lambda(w_t) = \frac{1}{b} \sum_{i=t-i}^{tb} \frac{-y_i x_i}{1+e^{y_i w_t^T x_i}} + 2\lambda w_t = 0$$

تقسیم 1 استفاده کردیم

تقسیم 2 تقسیم داده ها و تقسیم

Suppose big data set 10,000,000 point

dimention: ~~10000~~ 10^6 1000

$O(nd)$

how can we breeak this:

$w_0 = 0$ $\quad w_{t+1} = w_t - \gamma_t \nabla \hat{\mathcal{E}}_\lambda(w_t)$

if have 10000000 point in 1000 dimantion how much memory do you need?

when we are working in hurry

dataset big, not time to have iteration

if you don't have memory how you would implement?

Split it, I can't compute ∇ ---

Compute this for a part of data

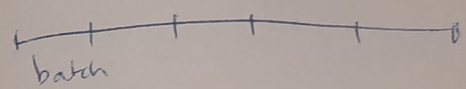
for b point

$$\nabla \hat{\mathcal{E}}_d(w_t) = \frac{1}{b} \sum_{i=1}^{b} \frac{-y_i x_i}{1+e^{y_i w_t^T x_i}} + 2\lambda w_t \neq 0$$

$g(t)$

compute a solution, exactly

must compute 1000 $\times$ 1000 $\times$ 1000



do γ update solution,

say: $w_t \rightarrow$ batch اول

$$\nabla \hat{\mathcal{E}}_\lambda(w_1) = \frac{1}{b} \sum_{i=1}^{b} \frac{-y_i x_i}{1+e^{y_i w_1^T x_i}} + 2\lambda w_1 \neq 0$$

for w_2: batch دوم

$$\nabla \hat{\mathcal{E}}_\lambda(w_2) = \frac{1}{b} \sum_{i=b+1}^{2b} \frac{-y_i x_i}{1+e^{y_i w_2^T x_i}} + 2\lambda w_2 = 0$$

$\longrightarrow$ so for t (برای t) batch = t

$$\nabla \hat{\mathcal{E}}_\lambda(w_t) = \frac{1}{b} \sum_{i=t-1}^{tb} \frac{-y_i x_i}{1+e^{y_i w_t^T x_i}} + 2\lambda w_t = 0$$

اشکال ۱ استفاده از این

اشکال ۲ : تکرار این داده ها روی تکرارات

T_{max} (τ)

threshold

γ_t

b

how to choose stuff

to compute *, ** must have all data

$\|w_t - w_{t+1}\|$ is cheap but *, ** are

cheap but it is easy to stop in chunks of data.

$\frac{n}{b}$ = epoch

talk about iteration

number of recieve data (epoch)

I am not changing the?

main issue when decend:

how invented Newtons

$\gamma_t = \frac{const}{\sqrt{t}}$ ← however / very important

mini batches

$\|\nabla F(w_t)\|^2 \gamma \tau^2$?

I split data,

⇐ mini randmiz—

socastic ?

how can change order

Permutation Version

randomly shuffle data,

when get to the end, shuffle again, sampling with replacement, sampling all of them & put it back

very so Catastic Vection

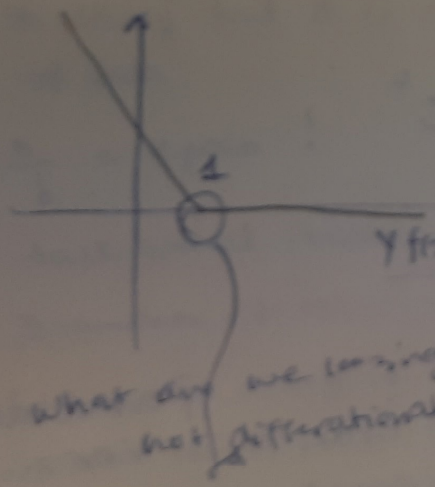
select 1 point of bigger Y put it back, 1 point of ..... it not repeated, after a epoch, ...

what you should get:

SVM:

is another loss function

Violet like



what are we losing, still convex but not differentiable

Can not do the gradients

if you do not have derivative but left derivative, you have angle, ...

everything set in left derivation

$$|1 - y w^T x|_+ = \max\{0, 1 - y w^T x\} \quad *$$

(zero if below zero)

when $y w^T x$ is > 0 → zero

$y w^T x < 1$ → this is like with step -1

this * is expression of last function

just know

$$\min \sum_{i=1}^{n} |1 - y_i w^T x_i|_+ + \lambda \|w\|^2$$

$E\lambda(w)$

You can not take gradient anywhere

so $\partial E_\lambda(w_t|_+ = \frac{1}{n} \sum_{i=1}^{n} - y_i x_i$ (left derivative) st

$y_i \cdot w^T x_i > 0$

$y_i w^T x_i < 0$

$$\text{sign}(x) \begin{cases} 0 \\ -1 \end{cases}$$

$w_0 = 0, \quad w_2 = w_1 - \gamma_t \nabla \hat{\xi}_\lambda(w_t)$

gradient decent,

ML Algorithmic Toolbox

LOCALITY KNN

close Input → same output

you avarage the y_i :

$$\hat{f}(x) = \sum_{i \in I(x)} \frac{y_i}{K}$$

there are afew computation:

what you do to calculate I(x)

1- sort
2- find K

the main problem: local

good → you can adapt simple function (line)

we look at linear models

what are the algorithm you know

RLS,

What is RLS:

just tell: best line to computed least square error.

~~You add~~. Take a line

check if good or not

if good

sum them up

& try to search to find min

$$\min_{w \in \mathbb{R}^d} \sum_{i=1}^{n} (w^T x_i - y_i)^2 + \lambda \|w\|^2 \quad \lambda \geq 0$$

do you need to λ equal to zero?

if λ is small you have data

if λ is big: how decide it is?

Check & try different λ & compute solution & check error, and check error on training set. the

? test

how to compute:

gradian.

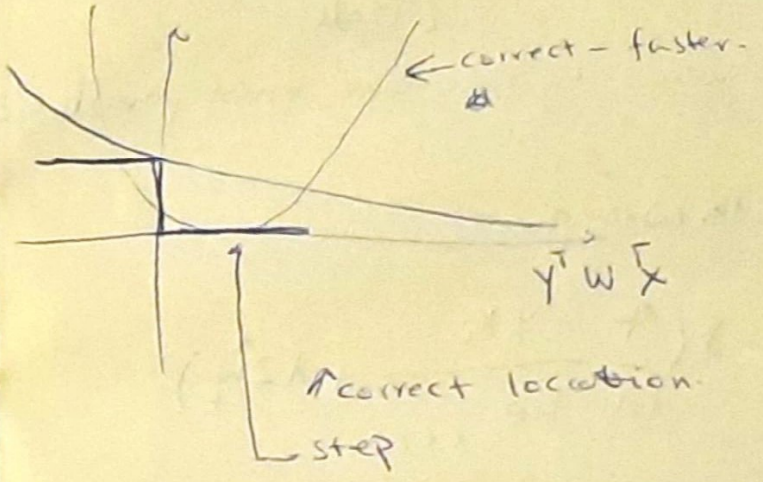
to find min & set equal to zero

& find a linear equation

$$\hat{w} = (\hat{X}^T \hat{X} + \lambda I)^{-1} \hat{X}^T y$$

RLS: logistic Regression

minimize $\log(1 + e^{-y_i w^T x_i})$



problem: nighter convex & nor differentiable

differentiable

mostly

in regression & classification

$$\sum_{i=1}^{n} (w^T x_i - y_i)^2 + \lambda \|w\|^2 \quad \lambda \geq 0$$

in classification:

$$\sum_{i=1}^{n} \log(1 + e^{-y_i w^T x_i}) + \lambda \|w\|^2 \quad \lambda \geq 0$$

$$\hat{w}_{t+1} = \hat{w}_t - \gamma X^T (\hat{X} w_T - \hat{Y})$$

$O(mdt)$ — ~~replace by~~ SVD — (Size)

if you want to calculate must parallel it.

X sth wrong.

$$\hat{w}_{t+1} = \hat{w}_t - \gamma \left(\sum_{i=1}^{n} \frac{-y_i x_i}{1+e^{x \times ??}} + \lambda 2 \hat{w}_t \right)$$

$$\text{ERM} \cdot \min_{w \in \mathbb{R}^d} \sum_{i=1}^{n} \ell(y_i, w^T x) + \lambda \|w\|^2$$

SGD

$$\hat{w}_{t+1} = \hat{K}_t - \gamma_t (\nabla \ell(y_{it}, \hat{w}_t^T x_{it}) + 2\lambda \hat{w}_t$$

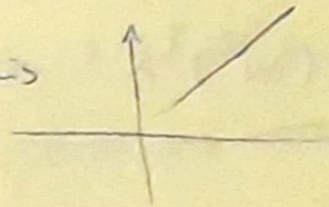
$f(x) = w^T x$

function which are linear

there are 2 ways

what non linear means

instead of this



I have wavie

linearity : sum , γ multiply by numbe

we have w γ x , we have to prove linearity for x γ w

- non linear :

sin, abs,
cos, square

make the formula not linear

So we have many option to do non linear

$f(x) = w^T x$

( ) $\in \mathbb{R}$

1. $f(x) = w^T (x)^2$ ← respect to x non
2. $f(x) = (w^T x)^2$ → both, non linear
3. $f(x) = (w^2)^T x$ → w non linear

You have to code,
instead of using line, I want to
~~use~~ use polynomial y ....
no additional code
we have $x_1 - x \in \mathbb{R}$
$f(x) = w^T x$

I ask to do $f(x) = ax^2 + bx + c$
invent a way to recycle the code you have to make this

$RLS(\hat{x}, \hat{y}, \lambda) = \hat{w}_x$

use the same code fit the
$f(x) = ax^2 + bx + c$
use RLS code & find a, b, c

(variable just changed)
$f(z) = w^T z$ $\hat{w}_x = RLS(\hat{z}, \hat{y}, \lambda)$

idea is:

$f(x) = ax + b$
$w(a, b)$, $z = (x, 1)$

new $w = (a, b, c)$ $z \neq (x^2, x, 1)$

fit polynomial in some sense is
line in z, linear combination
of monomials, its a line up
to change value,
$w(a, b, c)$, $z = (x^2, x, 1)$

$\hat{Z}$ = feature - map ($\hat{X}$)

fore any input, create corresponding Z

How to write polonomial degree Three

number of parameter in model ↑ p

$f(x) = ax^3 + bx^2 + cx + d$

$W = (a, b, c, d)$ $\quad Z = (x^3, x^2, x, 1) \in \mathbb{R}$

$\hat{Z}$ = Feature - map($\hat{X}$)

we want to see how powerfull the

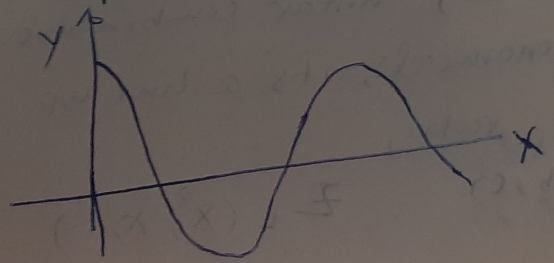
trick is;

this allowed as to go from polonomi

all to a linear form.

Even in 1 dimanation, Z changed.

Sappose that we have this functio



if you need polonomial, ~~which~~ &
Sin.

$f(x) = \cos(x)$

$A\cos(x)$

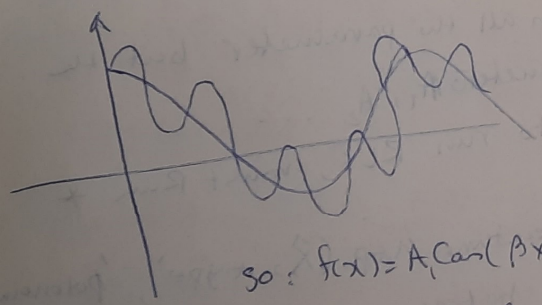
or we can have amplitute or

frequency

$f(x) = A\cos(\beta x)$

we have sin which modulated by

faster sin or cosi



so: $f(x) = A_1\cos(\beta x) +$

$A_2\cos(\beta_2 x)$

This function, can we wite it

with $f(z) = w^T z$, so:

$W = (A, B, A_2, B_2)$

$Z =$

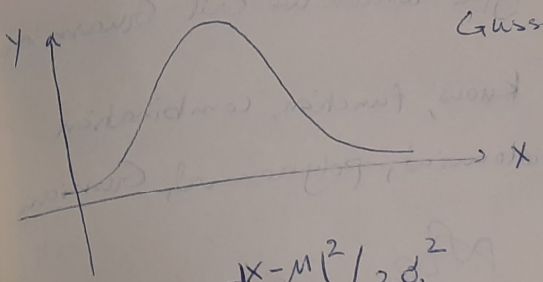
this trick does not work, for B
we cannot find β. if we want
to do this :

$$W = (A_1, A_2)$$

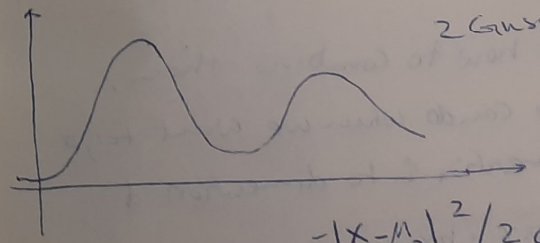
$$Z = (Con(\beta_1 X), Con(\beta X))$$

non all the parameter but the
parameters $A_1, A_2 \ldots$
to run RLS, must Run *

$\hat{Z}$ = Feature-Map ($\hat{X}$, 'type', 'polonomial')
as same before



$$f(x) = c_1 e^{-|x-\mu_1|^2 / 2\sigma_1^2}$$



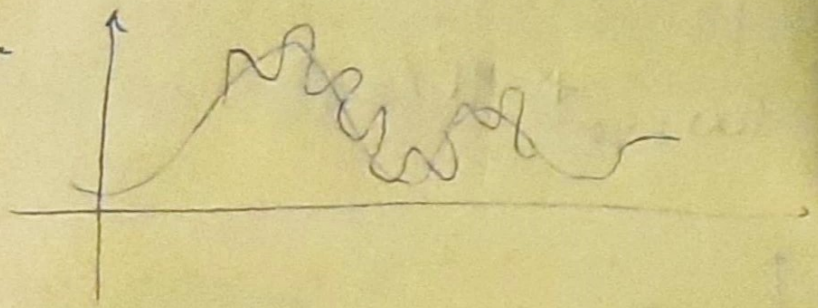
$$f(x) = c_2 e^{-|x-\mu_2|^2 / 2\sigma_2^2}$$

the parameter to find $c_1, c_2, \mu_1, \mu_2, \sigma_1, \sigma_2$

~~Can I do~~
we can find W, no because only c_1, c_2
are linear, instead of pick 2, pick
many, put over place & combine it.

Another type which we call Guassian

How you know, function, combination of monolomial, polynomial, Gussian

Like 

decide how to combine them,

What we can do when we want to go from dimention 1 to dimention d:

$f(x) = ax + bx + c$ → 2 dimention

$X : (x_1^1, x_2^2)$

$f(x) = a(x_1^1)^2 + b(x^2)^2 + c\, x^1 x^2$

same trick just the same,

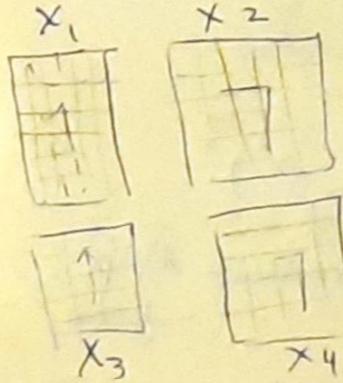
$f(x) = A_1 C_{in}(\beta_1^T x) + A_2 C_{in}(\beta_2^T x)$

$f(x) = c_1 e^{-\|x_i - M_1\|^2 / 2\sigma_1^2}$

$f(x) = c_2 e^{-\|x - M_2\|^2 / 2\sigma_2^2}$

think always > bg over parameterize.

You have 2 images

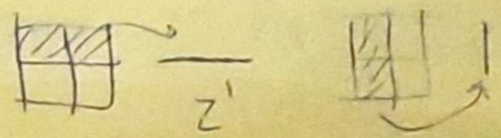


Can we use least square to solve proble

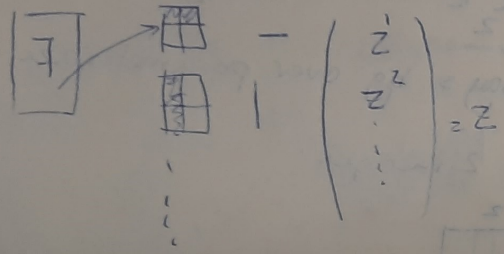
take regression & run it

Pixels are not very good, when you treat as a vector ? ? ?

take windows & take differences, because there is difference that change



So, ~~in~~ we built z, and bunch of number you obtain by manipulating $\hat{x}$



$\hat{z}$ = Feature_map($\hat{x}$, 'typ', 'p')

$\Phi : X \longrightarrow Z$ ⇝ feature Map

$\Phi(x) = \begin{pmatrix} f_1(x) \\ \vdots \\ f_p(x) \end{pmatrix}$: called features

$f_j : X \to \mathbb{R} \quad j = 1 \cdots p$

this is just abstraction,

Give me three & fours of f

f ~~must~~ could be sin, cos, polonimial, Gaussian, mixed of two,

notice that:

everything we disguss:

Whenever you see X,

z or $\Phi(x)$ are the same

$X \to z = \Phi(x)$

~~wrong~~

Feature MAPS $\longmapsto$ KERNELS

$$\min_{w \in \mathbb{R}^d} \sum_{i=1}^{n} \ell(x_i^T w, y_i) + \lambda \|w\|^2$$

$\Phi(x)^T w$

You can use linear parametrization to do nolinear not depend on w

this model is parameterize linearly
but it not linear.
We talk about w $\forall x$.

$\phi(x)^T w$

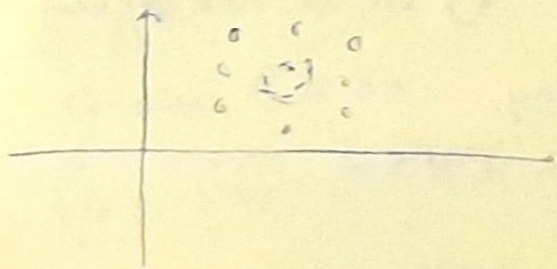
the beauty is it's a line
what are example of ϕ
Sign, abs, exponential,
whatever given by list of feature
$\phi(x) = (f_1(x) \dots f_p(x))$, $f_j : X \to \mathbb{R}$
~~anytime you can~~

1, X, has to be vector
if x is graph, the set of vertisiz
and ages, can we define the model?
No

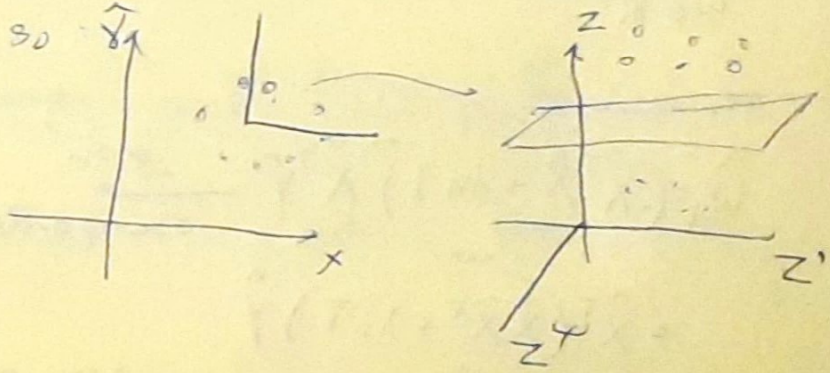
what are some number, classify people
each person is graph. ⊕ ⊗ we can
try $f(x)$: degree of graph

you can use string
data specific : ϕ
$\forall$ also be sth else, in situation not
vector structured
new idea. the moment choose ϕ (not know
how)

how do you use it.
last time called $z = \phi(x) = (f_1(x) \dots f_p(x))$



if you can use linear, for clarity,
~~not~~ can not because think of a plate



~~$\phi_{(x)} = Bx$~~

~~$W^T Bx = (B^T W)^T$~~

if ϕ is nonlinear,

How to ~~know~~ ϕ or linear.

I not know

main point : cross validation,
try on λ to get best

purpose of today
reparameterize problem
and infinit:

$$\min_{w \in \mathbb{R}^p} \sum_{i=1}^{n} l\left(\phi(x_i)^T w, y_i\right) + \lambda \|w\|^2 \quad \text{I}$$

remember:

$$\hat{w}_\lambda = \underbrace{(\hat{X}^T \hat{X}}_{m \times d} + \lambda n I)^{-1} \underbrace{\hat{X}^T \hat{Y}}_{d \times m} \xrightarrow[\text{swap order}]{\text{or}}$$

$$= \hat{X}^T \underbrace{(\hat{X}\hat{X}^T + \lambda n I)}_{n \times n} \hat{Y}$$

$d \times n$ $\quad n \times d$ $\quad d \times n$

give name ~~$\in \mathbb{R}^n$~~, $c = (\hat{X}\hat{X}^T + n\lambda I)^{-1} \hat{Y}$

SVD

$\hat{X} = UDV^T$

$\hat{X}^T = VDU^T$

~~skip and prove~~

finish the collection to $Q(x)$

$\hat{w}_\lambda = \hat{X}^T c = \sum_{i=1}^{n} x_i c_i$ (correspond to $\hat{X}$) $\quad c = (\hat{X}\hat{X}^T + n\lambda I)^{-1} \hat{Y}$

$\hat{f}_\lambda(x) = x^T \hat{w}_\lambda$ $\quad$ **

Compine * & **

$\hat{f}_\lambda(x) = x^T \hat{w}_\lambda = \sum_{i=1}^{n} x^T x_i c_i$

~~who~~ we want to replace x with $\phi(x)$ in $\hat{\phi}_{n \times p}$

$\hat{w}_\lambda =$

tediously: rewrite just put $\hat{X} \to \hat{\phi}$, $x \to \hat{\phi}$

just:

$$\hat{w}_\lambda = (\hat{\Phi}^T \hat{\Phi}_{n\times p} + \lambda n I)^{-1} \Phi^T \hat{Y}$$

$$(\cdots) = \sum_{i=1}^{n} \Phi(x_i) c_i \quad , c = (\hat{\Phi}\hat{\Phi}^T + n\lambda I)^{-1} \hat{Y}$$

$$\hat{f}_\lambda(x_1) = \sum_{i=1}^{n} \Phi(x_1)^T \Phi(x_i) c_i$$

so just change notation.

1/ observation: if $\hat{w}_\lambda$, but $\hat{f}(w)$ just look at $\hat{f}$ just means look at prediction of w, what about c, how compute what is the entry of $\hat{\Phi}\hat{\Phi}^T$

matrix

$$(\hat{\Phi}\hat{\Phi}^T)_{ij} = \Phi(x_i)^T \Phi(x_j)$$

inner product between 2 vectors

$$\hat{X} = \begin{pmatrix} x \\ \vdots \\ x \end{pmatrix}^{n\times d}, \quad \hat{\Phi} = \begin{pmatrix} \Phi(x_1) \\ \vdots \\ \Phi(x_n) \end{pmatrix}^{n\times p}$$

$$\Phi(x)^T \Phi(x') = \sum_{j=1}^{p} f_j(x) f_j(x') = K(x, x')$$

give a name

inner product of any x,

question can we imagine to compute this experation when $j \to +\infty$

answer of do, doesn't pay attention?

Geomatric series

$$\sum_{j=0}^{\infty} a^j = \frac{1}{1-a} \qquad 0 < a < 1$$

Exponetial series

$$\sum_{j=0}^{\infty} \frac{a^j}{j!} = e^a$$

we want to use these 2 to invent, write down the K.

~~1-1~~

$X = \{0, 1\}$

not only one dimentional

invent ϕ_j

to do the sum

$$\phi(x)^T \phi(x') = \sum_{j \geq 0}^{\infty} \phi_j(x) \phi_j(x') = K(x, x')$$

consider $\phi_j(x) = x^j \quad j = 0, \ldots \infty$

$$\sum_{j \geq 0}^{\infty} x^j x'^j = \sum_{j \geq 0}^{\infty} (x x')^j = \frac{1}{1 - x x'} \sum_{j \geq 0}^{\infty} w_j x^j$$

$$f(x) = \sum_{j \geq 0}^{\infty} w_j \phi_j(x) =$$

~~if~~ example if $X(0, C)$

$$\phi_j(x) = \left(\frac{x}{\sqrt{C}}\right)^j \quad j = 0, 1 \ldots \infty$$

$$\sum_{j \geq 0}^{\infty} \left(\frac{x}{\sqrt{C}}\right)^j \left(\frac{x'}{\sqrt{C}}\right)^j = \sum_{j \geq 0}^{\infty} \left(\frac{x x'}{C}\right)^j = \frac{1}{1 - \frac{x x'}{C}} = K(x, x') \qquad \sum_{j \geq 0}^{\infty} w_j x^j$$

Kernel is easy

but look wierd

ok for Exponential series:

$$K(x, x') = e^{-(x - x')^2 / 2\sigma^2} \quad \text{gaussian}$$

$=$

lets develop square:

$$K(x, x') = e^{-(x - x')^2 / 2\sigma^2} = e^{\frac{-x^2}{2\sigma^2}} e^{\frac{-x'^2}{2\sigma^2}} e^{\frac{2xx'}{2\sigma^2}} =$$

$$e^{\frac{-x^2}{2\sigma^2}} e^{\frac{-x'^2}{2\sigma^2}} \sum_{j \geq 0}^{\infty} \frac{1}{j!} \underbrace{\left(\frac{2xx'}{2\sigma^2}\right)^j}_{\text{we have a and}}$$

need to massage to find ϕ_j

how can

$$= e^{\frac{-x^2}{2\sigma^2}} e^{\frac{-x'^2}{2\sigma^2}} e^{\frac{2xx'}{2\sigma^2}} = e^{\frac{-x^2}{2\sigma^2}} e^{\frac{-x'^2}{2\sigma^2}} \sum_{j \geq 0}^{\infty} \frac{1}{j!} \left(\frac{2}{2\sigma^2} xx'\right)^j$$

$$= \sum_{j \geq 0}^{\infty} \left(\frac{\sqrt{2} x^j e^{\frac{-x^2}{2\sigma^2}}}{\sqrt{2} \sigma^j \sqrt{j!}}\right) \left(\frac{\sqrt{2}}{\sqrt{2}} \frac{x'^j}{\sigma^j} \frac{e^{\frac{-x'^2}{2\sigma^2}}}{\sigma \sqrt{j!}}\right)$$

$$= \sum_{j \geq 0}^{\infty} \phi_j(x) \qquad \phi_j(x')$$

Select the relevant expression to computable code: if I give you K

ϕ is ∞

$$f(x) = \sum_{(\dots)^{i\geq 1}}^{n} \phi(x)^T \phi(x_i)\, c_i$$

$$c = (\hat{\phi}\hat{\phi}^T + n\lambda I)^{-1} \hat{y} \rightarrow$$

$$(\hat{\phi}\hat{\phi}^T)_{n\times n}^{ij} = \phi(x_i)^T \phi(x_j)$$

$$\Rightarrow$$

$$y = \hat{f}_{\lambda}(x) = \sum_{i\geq 1}^{n} K(x, x_i)\, c_i$$

$$c = (\hat{K} + \lambda n I)^{-1} \hat{y}$$

$$\hat{K}_{ij} = K(x_i, x_j)$$

after turn these 3 expration to

Suedocode.

$c = KRLSI(\hat{x}, \hat{y}, \lambda, \underline{\text{'Kernel'}})$ call it ker_type

another function to call

$\hat{k} = kernel(\hat{x}, \underline{x}, \text{'ker\_typ'})$

↓ call it $\underline{k}$

$\Rightarrow c = KRLSI(\hat{x}\ \hat{y}\ \lambda, \text{'ker\_typ'})$

$\hat{k} = kerl(\hat{x}, \hat{x}, \dots)$

$$c = \frac{y}{(\hat{k} + n\lambda I)}$$

$K = Ker(\hat{x}, \underline{x}, \text{'kern\_typ'})$

$= KRLS\_test(c, x, \hat{x}, \text{'ker\_typ'})$

give new x

you have to compte

then inside can do 3 expration

for $i = 1 \dots n$

$\underline{K}(i) = Kernel(x x_i, \text{'type'})$

$y_{temp} = y_{temp} + K(i) c_i$

Neural networks

KNN is first Algorithm - nearest neighbours

local Algorithm.

depend on K, ~~we might have~~

smooth function, & wish to have linear function,

no constraints on the family of function for estimator $\hat{f}$.

if you have lots of data, ~~its~~

its local method,

1) number of data

2) $\tilde{=}$ 7, 18 nov

3) no hon of dimention ~~not~~ increase number of data

if dimention is big, distance become very large

Regularization methods - global (or Network) — RLS, logistic R with linear f

looking for estimator, introduce RLS

we want to minimize impencal Risk.

$$\hat{f} = \underset{f \in H}{\operatorname{argmin}} \left(\frac{1}{n} \sum_{i=1}^{n} \ell(y_i, f(x_i)) + \lambda \|f\|^2 \right)$$

regularization term

$f \in H$: family of function,

this has been general.

RLS (RLS with linear function)

we made specific least square, square ~~distance~~ loss

first choice:

$$\frac{1}{n} \sum_{i=1}^{n} (y_i - f(x_i))^2 + \lambda \|f\|^2$$

H: is class of linear function →

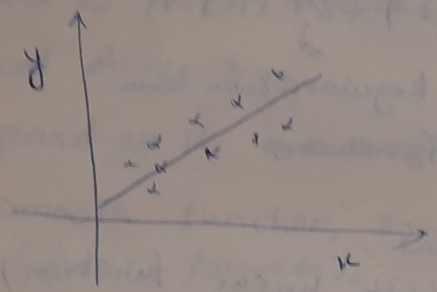
$$\frac{1}{n} \sum_{i=1}^{n} (y_i - w^T x_i)^2 + \lambda \|w\|^2$$

$\vec{w} = \underset{w \in \mathbb{R}^D}{\operatorname{argmin}}$

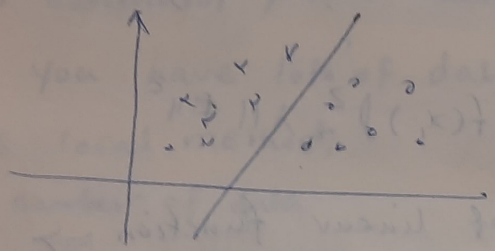
D: Dimention of x

LR logistic function

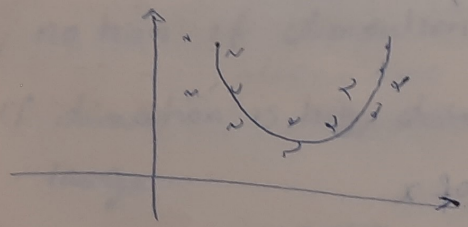
for regression



in binary classificatio

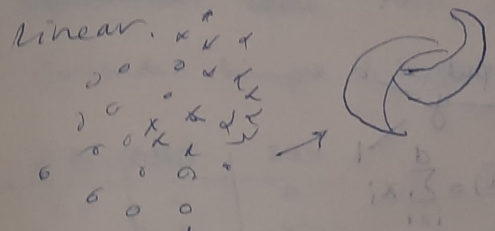


Non linear



feature map

if problem is not linear so increase dimention d, I can map point to higher dimentional space that are each pant is linear.



$$\Phi : \mathbb{R}^{d} \to \mathbb{R}^{p} \qquad p >> d$$

replace x with $\Phi(x)$

$$\hat{w} = \underset{w \in \mathbb{R}^p}{\operatorname{argmin}} \frac{1}{n} \sum_{i=1}^{n} (y_i - w^T \Phi(x_i))^2 + \lambda \|w\|^2$$

if K is Kernel function:

$$K(x_i, x_j) = \Phi(x_i)\,\Phi(x_j)$$

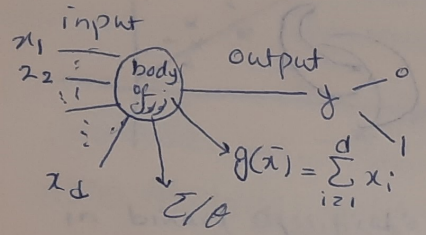
(27:15"

You can describe solution w as waited combination, with term, ~~comp~~

Neural Networks:

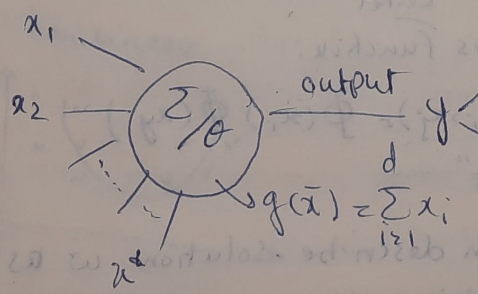
first model (1943) (single layer perceptron ~~proce~~)

main element: artificial neuron:



this artificial nuron accept vector $\bar{x} \in \mathbb{R}^d$

$$f(\bar{x}) = \begin{cases} 1 & \text{if } g(\bar{x}) > \theta \\ 0 & \text{otherwise} \end{cases}$$



all input have same important.

~~one connect~~

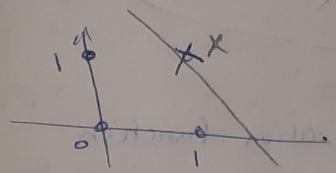
Complexity of model:

~~structure~~ ~~with~~ AND, OR, XOR, ...

Consider simple situation:

A B	AND	sum
0 0	0	0
0 1	0	1
1 0	0	1
1 1	1	2

try to use 2 input on plate:



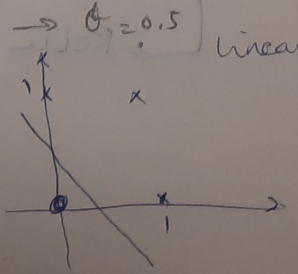
linear

We would like to find seperator like /

$\theta = 1.5$

A B	OR
0 0	0
0 1	1
1 0	1
1 1	1

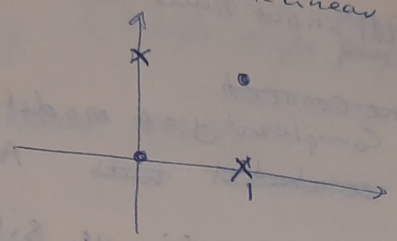
→ between (0,1)



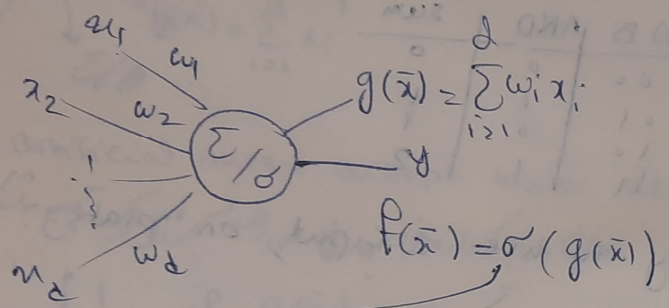
linear

A	B	XOR
0	0	0
0	1	1
1	0	1
1	1	0

non-linear

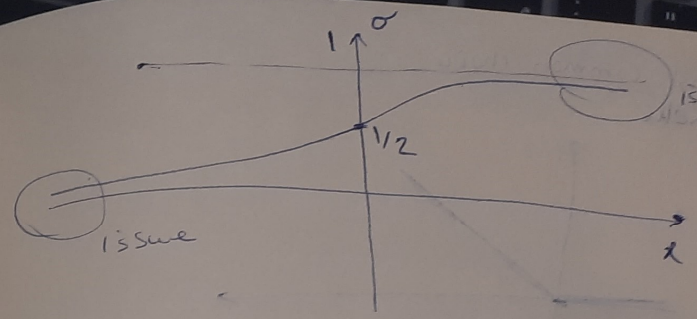


put weight



→ non linear activation function.

- sigmoid σ
- tanh
- Relu



$$\sigma(x) = \frac{1}{1+e^{-x}}$$

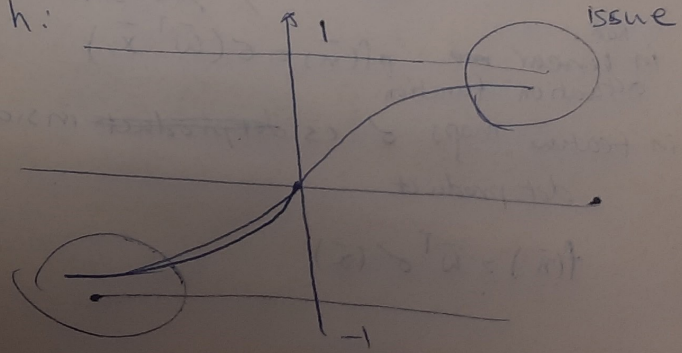
for any x, output is in Range $(0,1)$
easy to understand

for $x < 0$

$x > 0$

we all need to optimise the weight
need of computing gradient

tanh:



most common choice

Relu

$$Relu(x) = \max(0, x)$$

If ~~you~~

$$f(\bar{x}) = \sigma(\bar{\omega}^T \bar{x})$$

We can use feature maps & Kernels

In case of feature map:

Feature map:

$$f(\bar{x}) = \bar{\omega}^T \sigma(\bar{x})$$

↓

non linear model

linear model

non linear ~~no~~ activation function $pf(\bar{x}) = \sigma(\bar{\omega}^T \bar{x})$

in feature maps σ is ~~dot product~~ inside

- dot product

$$f(\bar{x}) = \bar{\omega}^T \sigma(\bar{x})$$

in linear model, extra term, which is Bias

call it ω_0 in our model

$$f(x) = \omega x + \omega_0$$

one thing: $f(x) = \omega x + \omega_0$

re write our function: $f(\bar{x}) = \sigma(\bar{\omega}^T [1, \bar{x}])$

$\bar{\omega} = [\omega_0, \omega_1, \dots \omega_d]$

~~in our model we have extra~~

take data & standardize with respect of min

trick : solution goes through origin

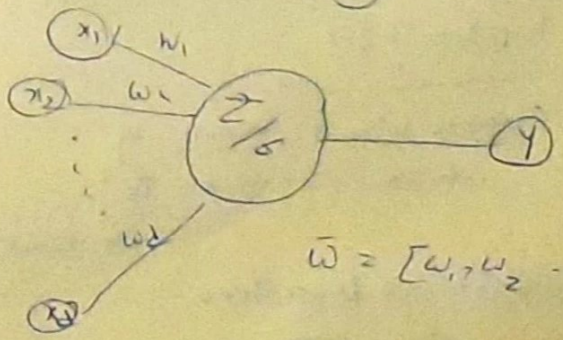
You have it.

we have a model

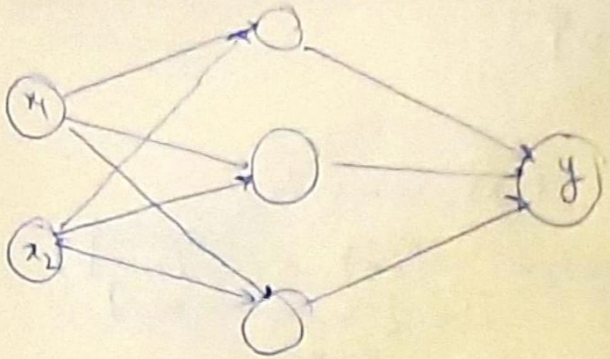
& start to stack norons together

Multi-layer perceptron (dense network

fully connected network)



$\bar{\omega} = [\omega_1, \omega_2 \dots \omega_d]$

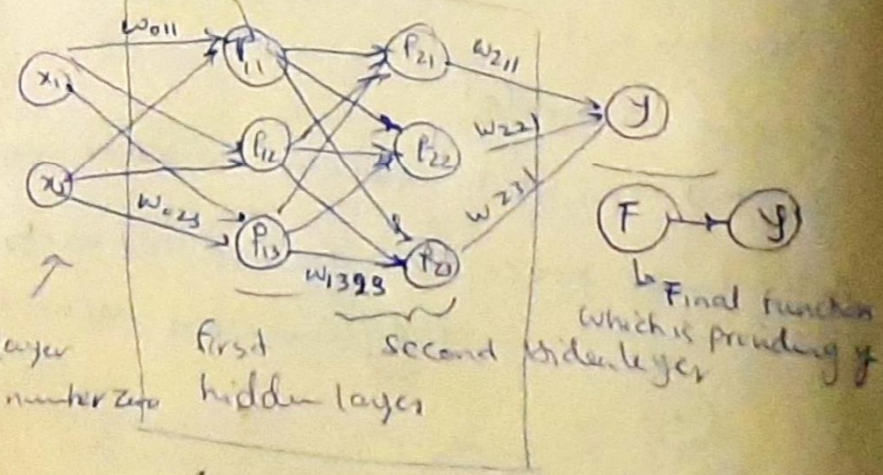


Input layer Hidden layer

↓

Not accessible from outside

first layer, first neron Single layer procaptron



Final function which is providing y

Layer number zero first hidden layer second hidden layer

Supervised learning which connect $\mathbb{R}^2$ in input to $\mathbb{R}$ output

w_{023} weight from layer Zero from neron 2 to neron 3

each neron which is hidden is like the first neron of the layer.

$$f_{11}(\bar{x}) = \sigma\left(\sum_{i=1}^{2} \omega_{oi1} x_i\right)$$

$$\bar{x} = [x_1, x_2]$$

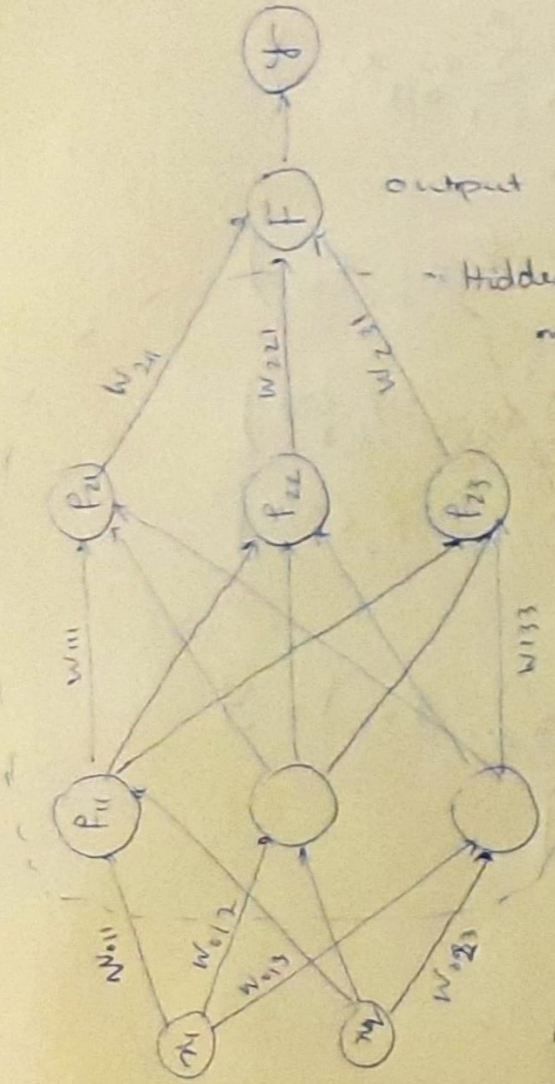
$$f_{12}(\bar{x}) = \sigma\left(\sum \omega_{oi2} x_i\right)$$

$$f_{23}(\bar{f}_1) = \sigma\left(\sum_{i=1}^{3} w_{1i3} \cdot f_{1i}^{(\bar{x})}\right)$$

$$\bar{f}_1 = \left[f_{11}(\bar{x}), f_{12}(\bar{x}), f_{13}(\bar{x})\right]$$

$$F(\bar{f}_2) = \sigma\left(\sum_{i=1}^{3} w_{2i1} f_{2i}\right) = y$$

$$\bar{f}_2 = [f_{21}, f_{22}, f_{23}]$$



$\hat{F}: X \to Y$

$X \subseteq \mathbb{R}^2$ → in this example

in General $\mathbb{R}^D$

$Y \subseteq \mathbb{R}$

We need to train:

initialize network with random weight

$F_{\bar{w}_0}$: initialization of weight

$F_{\bar{w}_0}(x_i) = \hat{y}_i \qquad \forall_i = 1 \dots n$

~~We can do the same without~~

we must compute loss. Loss would be the Loss we had before

it means that, estimate empirical Risk

Forward propagation

$$\frac{1}{n} \sum_{i=1}^{n} (y_i - F_{\bar{w}}(x_i))^2$$

$$\hat{w} = \underset{w \in \mathbb{R}^m}{\operatorname{argmin}}$$

in Regularized network

$$\hat{w} = \underset{w \in \mathbb{R}^m}{\operatorname{argmin}} \frac{1}{n} \sum_{i=1}^{n} (y_i - F_w(x_i))^2 + \lambda \|w\|^2$$

do not activation function

only parameter according to weight

with more neurons or more layers number of parameter to learn is explode

if number of parameter m is suppose high with respect to complexity of problem & data set is poor, you end up with overfitting.

* training is complicate, a good idea to balance the architecture

$$\hat{W} = \underset{W \in \mathbb{R}^m}{\operatorname{argmin}} \boxed{\frac{1}{n} \sum_{i=1}^{n} (y_i - F_W(x_i))^2} \; J(W)$$

Solve:

gradient = 0
respect to W

but it's hard, may be we have a lot of min locally.

there is advantage which is called chain rule of derivation to compute gradient

$$F(x) = f(g(h(x))), \quad \frac{dF}{dx} = \frac{dF}{dg} \cdot \frac{dg}{dh} \cdot \frac{dh}{dx}$$

$$\frac{\partial J(W)}{\partial W_K}$$

$K = 1 \dots m$

in training we have data x with Forward propagation & try & minimize each problem with $\frac{\partial J(W)}{\partial W_K}$, $K = 1 \dots m$

With partial derivative, is it possible to easy write down explicit solution: $\hat{W} = \dots$

not because of complexity,

W_0

$$W_t = W_{t-1} - \gamma \nabla J(W_{t-1})$$

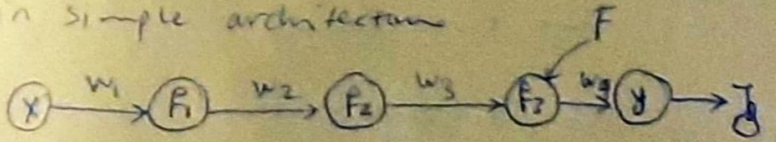
γ → fixed as Logistic Regression

multiplant the gradient of function

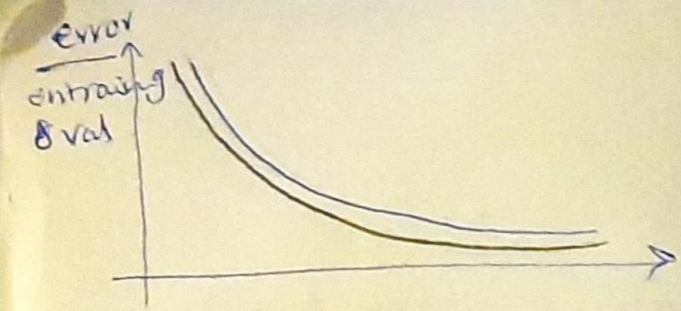
Back propagation

refer to process, compute the root & back

in simple architecture



back propagation



- train training

training —

validation —

Gradient descent

entire trainingset

stochastic GD

basis of single sample of traing & compute grad only according to that point

(the ~~path~~ path is unstable but close to min)

in logistic regression,

use Entire traing

Compute gradient

&

Conclude an epoch

for each epoch we need 1 epoch

a usual choice is not Gradient or stochastic

~~even~~ in very large dataset is ??

there is intermediate solution

not entire traing & not 1

subset of traingset

⇓

mini batch GD

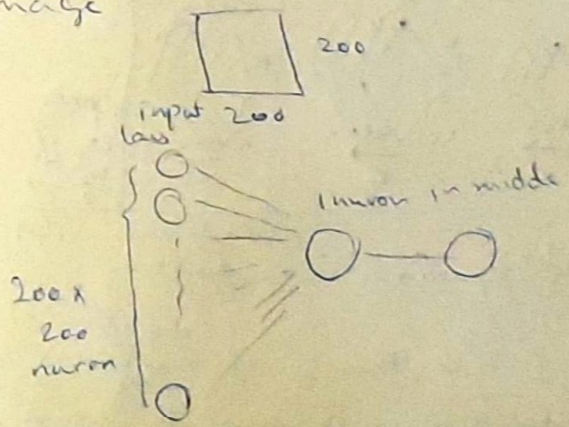
Enough subset to cover entire traingset

Introducing Convolutional Neural network

input data are characterize by topology.

we have,

image



2 observation

when you have image Y consider what happen in 1 of the pixel,

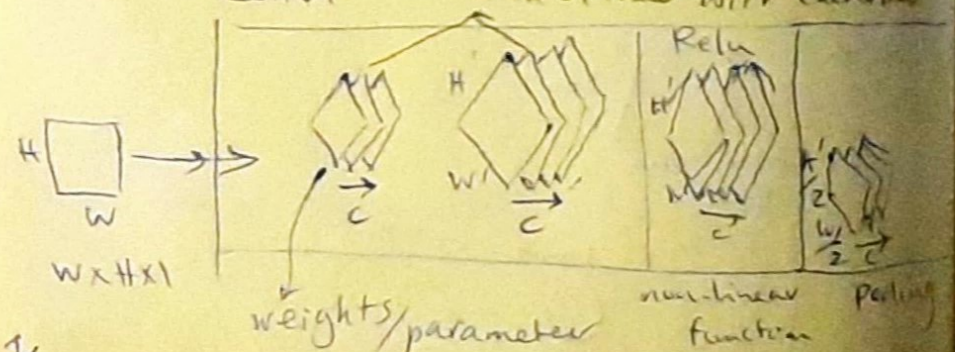
21.47 : 20 Nov

~~image~~

change to visualize:

H 200  200 W would turn to many input of neuron, this image provide layer which is called convolutional layer

convolutional layer has different steps we use filter by size K×K, result of applying is another visual Representation W' & H' (depend on size of filter



1/ Source interacho if I take this set of weight neuron & apply to different position of image.

2/ second: parameter sharing,

apply very safe parameter to image.

input neuron to some part of the image each neuron in input have own set of weight, ... ? everything is independent

3/ How many parameter would you have ~~so poor~~ in convolution layer, have more filter (with 1 its poor)

⇒ in representation have diffrent representation

How many parameters:

K × K × C

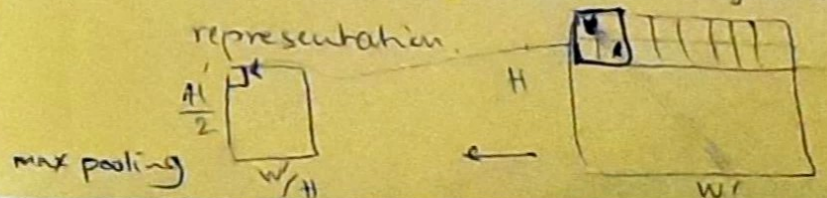
~~Having & initialize, refine to~~

second step: non linear activation function apply Relu to each element of W/H'

↳ doesn't change the size, remain the same

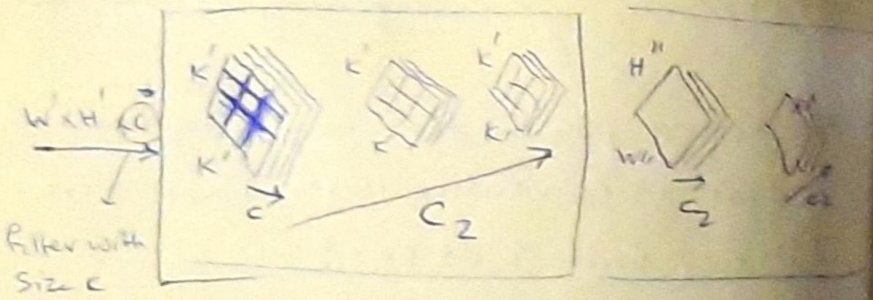
pooling: avoid if you had 2×2 patch.

doesn't change the strength of representation.

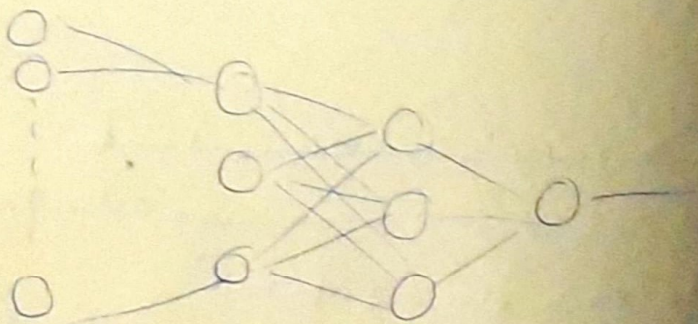


only parameters have to learn are the ones of convolution. the Convolution layer, play role in layer

$Conv_2$



in $Conv_2$ same as $Conv_1$



Number of

Title: Machine Learning 25 Nov

$P(X,Y)$ $\quad f: X \to Y \quad \mathbb{E}_{(X,Y)\sim P}[\ell(Y, f(X))] =$

ERM $\quad \frac{1}{n}\sum_{i=1}^{n} \ell(y_i, f_\theta(x_i)) + \lambda R(\theta)$ $\quad$ Regularization

$\|\theta\|^2$

Regularization

1 FREQUENTISTS

2 DISCRIMINATIVE : take decision in $P(X,Y)$

4 GENERATIVE / PROBABILISTIC

3 BAYESIAN

Probability:

$P_\theta(X,Y)$: parametrized, function would be parametrized θ.

Typically,

$p(x,y) = P_\theta(x,y)$

we are going to MLE (Maximum Likelyhood), MAP (BAYESIAN LEARNING)

How we can estimate mean in distribution

~~$R \to Z \sim p(x,y)$~~

$R \to Z \sim P(Z)$, $M = \mathbb{E}[Z] = \frac{1}{n}\sum_{i=1}^{n} z_i$ $(z_1 \dots z_n)$ estimate mean

Why it is good: the estimator is on Bias

$\mathbb{E}[\hat{M}] = M$ $\quad P(|\hat{M} - M| > \varepsilon) \to 0$ $\quad n \to \infty$

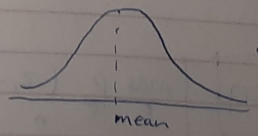
I want to find P_{θ_*} ~~not~~

$p(z) = P_{\theta_*}(z)$ $\quad z_1 \dots z_n \sim P_{\theta_*}$ $\quad$ Date.

example of ~~probability~~ $P_\theta(z) = \frac{e^{-|z-\mu|^2/2\sigma^2}}{\sqrt{2\pi}\,\sigma}$

$\theta = (\mu, \sigma)$

distribution: (mean & variance)



← could be linear

Guassian distribution.

How determine models:

$P(z) = P_{\theta_*}(z)$, $z_1 \dots z_n \sim P^n_{\theta_*}$

which θ generated my θ $\quad \theta \to z_1^{\theta_1} \dots z_n^{\theta_1} \sim P_{\theta_1}$ $\quad \theta_2 \dots$

Opponent

what is probability of data under this distribution

θ_* : right one;

$P^n_{\theta_*} = P_{\theta_*}(z_1 \dots z_n) = \prod_{i=1}^{n} P_{\theta_n}(z_i) \gtrless P_{\theta_1}(z_1 \dots z_n) = \prod_{i=1}^{n} P_\theta(z_i)$

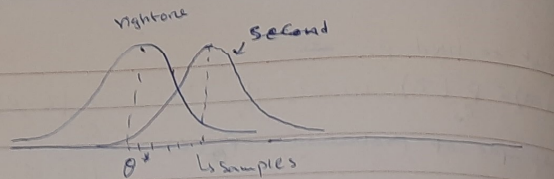
متوسط → متوسط (مقدار، میانگین، ...)

if data are gussian $\quad z_i = \mu \pm \sigma^2 / z, \theta_* = \mu_*$

$P_{\theta_*}(z_1 \dots z_n) = \prod_{i=1}^{n} \frac{1}{(\sqrt{2\pi})} e$

Delijia

Title.



if computer probibily but I can not find better than the right mass

$$P^m_{\theta^*} = P_{\theta^*}(z - z_n) = \prod_{i=1}^{n} P_{\theta^*}(z_i)$$

likely hood of θ

$$\max_{\theta} P_\theta(z_1 \dots z_n) \Leftrightarrow \max_{\theta} \prod_{i=1}^{n} P_{\theta_i}(z_i)$$

maximum likelihood estimation (MLE)

we want to chase probibily

$$\hat{L}(\theta) = P_\theta(z_1 - z_n) = \prod_{i=1}^{n} P_\theta(z_i)$$

$$\hat{L}(\theta) = P_\theta(z_1 \dots z_n) = \prod_{i=1}^{n} \frac{1}{(\sqrt{2}\pi)} e^{-|z_i - \mu^2|/2}$$

lets change the expuension with log:

log likely hood:

log of Product is like sum $\Rightarrow$ $\log \prod_{i=1}^{n} P_\theta(z_i) = \sum_{i=1}^{n} \log P_\theta(z_i)$

$$\log \hat{L}(\theta) = \sum_{i=1}^{n} \left[\log\left(\frac{1}{\sqrt{2\pi}}\right) + \log e^{-|z_i - \mu|^2/2} \right] =$$

$$\log \hat{L}(\theta) = \sum_{i=1}^{n} \left(\log\left(\frac{1}{\sqrt{2\pi}}\right) + \left(-|z_i - \theta|^2/2\right) \right]$$

we call it θ

II

to maximize this expresson must maximize the II part

$$\min_{\theta} \sum_{i=1}^{n} \frac{|z_i - \theta|^2}{z_i}$$

$$-\sum_{i=1}^{n} \frac{2(z_i - \theta)}{2} = 0 \quad , \quad \sum_{i=1}^{n} \theta = \sum_{i=1}^{n} z_i$$

$$\theta_n, \quad \hat{\theta} = \frac{1}{n} \sum_{i=1}^{n} z_i$$

$P(X,Y)$

$P_\theta(X,Y) = P_\theta(Y|X) P_\theta(X)$

Regression

make a model:

$Y_i = w_*^T x_i + \varepsilon_i$, $\varepsilon \sim \mathcal{N}(0, \sigma^2)$

↳ noise is guassion.

$Y_i \sim P_\theta(Y_i | X_i) = ? \mathcal{N}(w_*^T x_i, \sigma^2)$

X is fixed

$$\left[P_{\theta_*}(Y|X) = \mathcal{N}(\theta_*^T X, \sigma^2) = \frac{1}{c} e^{-|Y - \theta^T X|^2/2\sigma^2} \right.$$

σ^2 fixed given

$$\hat{L}(\theta) = \prod_{i=1}^{n} \frac{1}{c} e^{-|Y_i - \theta^T x_i|^2/2\sigma^2}$$

now take log:

$$\log \hat{L}(\theta) = \sum_{i=1}^{n} \left[\text{Constant} - (Y_i - \theta^T x_i)^2/2\sigma^2 \right]$$

$$\max \hat{L}(\theta) \Leftrightarrow \min_{\theta \in \mathbb{R}^d} \sum_{i=1}^{n} (Y_i - \theta^T x)^2/2\sigma^2$$

log

min least square

Delijia

Title. Classification $y \in \{\pm 1\}$

$$\begin{cases} P_{\theta_*}(y|x) = \sigma(y\theta_*^T x), \quad \sigma(z) = \frac{1}{1+e^{-z}} \\ P(1|x) = \frac{1}{1+e^{-y\theta^T x}} = \frac{1}{1+e^{-\theta^T x}} \end{cases}$$

$$P_\theta(1|x) = \frac{1}{1+e^{-\theta^T x}} \times \frac{e^{-\theta^T x}}{e^{-\theta^T x}} \Rightarrow$$

Logistic model ;

$$P_\theta(-1|x) = \frac{e^{-\theta^T x}}{1+e^{-\theta^T x}}$$

~~$\hat{L}(\theta|x) = \prod_{i=1}^{n}$~~

$$\frac{P_\theta(y_i|x_i)}{} \quad \Leftrightarrow P_{\theta_*}(y|x) = \frac{1}{e^{-y\theta_*^T x}}$$

$$\hat{L}(\theta) = \prod_{i=1}^{n} \frac{1}{1+e^{-y_i\theta^T x_i}}$$

$$\log \hat{L}(\theta) = \sum_{i=1}^{n} -\log(1+e^{-y_i\theta^T x_i})$$

$$\max_{\theta \in \mathbb{R}^d} \hat{L}(\theta) \Leftrightarrow \min \sum_{i=1}^{n} \log(1+e^{-y_i\theta^T x_i})$$

logistic regression when

$y \in (0,1)$

$$P(y|x) = \frac{1}{1+e^{-\theta^T x}}$$

$$\hat{\mu} = \frac{1}{n}\sum_{i=1}^{n} z_i$$

$$\hat{w} = \arg\min \sum_{i=1}^{n} (y_i - w^T x_i)^2$$

$$\hat{w} = \arg\min_{w \in \mathbb{R}^d} \sum_{i=1}^{n} \log(1+e^{-y_i w^T x_i})$$

MLE

Z (random value) $\mathbb{R}^d \ni Z \sim P_{\theta_*}(z) = P_*(z)$

distribution random value)

$(z_1 \cdots z_n) \sim (P_{\theta_*})^n$ (iid)

instead of finding P_*

$\hat{P} \sim P_* \Leftrightarrow \hat{\theta} \approx \theta_*$

$$\hat{L}(\theta) = \prod_{i=1}^{n} P_\theta(z_i)$$

$$\log \hat{L}(\theta) = \sum_{i=1}^{n} \log P_\theta(z_i)$$

because log is easy in write sum of product

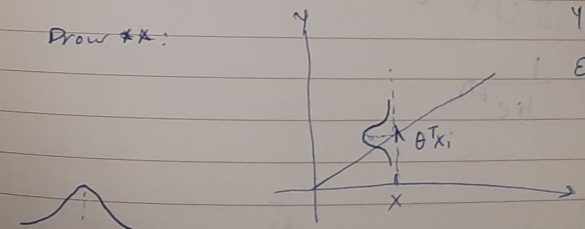
Maximize likelyhood, $\max \log \hat{L}(\theta)$

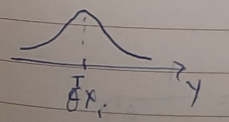
** $P_\theta(z) = \frac{1}{c} e^{-\|z-\theta\|^2/2\sigma^2}$ $\theta \in \mathbb{R}^p$ (σ^2 known), $\log \hat{L}(\theta) \propto -\sum_{i=1}^{n}(z_i-\theta)^2$

$$\hat{w} = \arg\min_{w \in \mathbb{R}^d} \sum_{i=1}^{n} (y_i - w^T x_i)^2 \quad P_\theta(y|x) \propto \frac{1}{c} e^{-\|y_i - \theta^T x_i\|^2/2\sigma^2}$$

$$\log \hat{L}(\theta) \propto -\sum_{i=1}^{n} (y_i - \theta^T x_i)^2$$

Draw **:

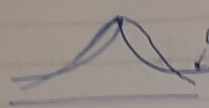




Delijia

Title

instead of Gaussian, use Laplacian. ($\mathcal{N}(\mu, \sigma^2)$)



$\frac{1}{c} e^{-\|z-\mu\|/2\sigma}$

What is difference between Gaussian & Laplacian

its a square

{ use absolute value when ...
{ use when

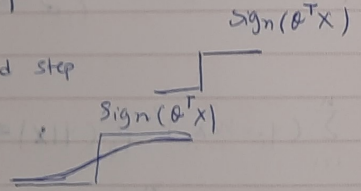
Laplacian is - Always

$$P_\theta(Y|X) = \frac{1}{1+e^{-Y_i \cdot \theta^T X_i}}$$

$Y \in \{\pm 1\}$

$P(1|X) + P(-1|X) = 1$

idea: shape: smoothed step



$$\text{Log } \hat{L}(\theta) = -\sum_{i=1}^{n} \log\left(1+e^{Y_i \theta^T X_i}\right)$$

$$P_\theta(1|X) = \frac{1}{1+e^{-\theta^T X_i}}$$

$Y \in \{0,1\}$

~~$\log \hat{L}(\theta) =$~~

$$P_\theta(0|X) = 1 - \frac{1}{1+e^{-\theta X}} = \frac{e^{-\theta^T X}}{1+e^{-\theta X}} = \frac{1}{1+e^{\theta^T X}}$$

$$P_\theta(Y|X) = \left(\frac{1}{1+e^{\theta^T X}}\right)^Y \left(\frac{1}{1+e^{\theta^T X}}\right)^{1-Y}$$

Date.

$$\log \hat{L}(\theta) = -\sum_{i=1}^{n} \left[Y_i \log\left(1+e^{-\theta^T X_i}\right) + (1-Y_i)\log\left(1+e^{\theta^T X_i}\right)\right]$$

: (Crosentropy).

$$\hat{w} = \underset{w \in \mathbb{R}^d}{\text{argmin}} \sum_{i=1}^{n} (Y_i - w^T X_i)^2 + \lambda \|w\|^2$$

$$\hat{w} = \underset{w \in \mathbb{R}^d}{\text{argmin}} \sum_{i=1}^{n} \log\left(1+e^{-Y_i w^T X_i}\right) + \lambda \|w\|^2$$

BAYES:

Z, U

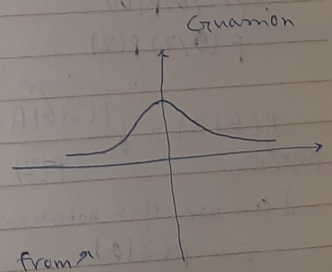
(2 random variable instead of X, Y)

$P(Z,U) \Rightarrow P(U|Z) = \frac{P(Z,U)}{P(Z)}$

$$P(Z|U) = \frac{P(Z,U)}{P(U)}$$

$$P(Z,U) = P(Z|U)P(U)$$
$$= P(U|Z)P(Z)$$

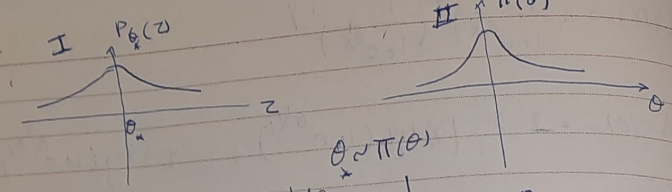
$$P(U|Z) = \frac{P(Z|U)P(U)}{P(Z)}$$



data comes from true model from

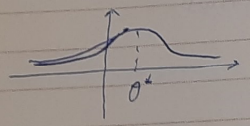
Delijia

Title.



how do I generate data,
First sample

Pick 1 guy (sample) from II, put on P_{θ_*} ✓
get distribution. (generate it according to I)



~~Sample acc.~~
$z \sim P_{\theta}(z)$

(data Generation)

PRIOR
can you learn it, No, Choice, it.

$p(z,\theta) = p(\theta,z) = \frac{p(z,\theta)}{p(z)}$

$= p(z/\theta)p(\theta)$ $\quad p(z|\theta) = \frac{p(z,\theta)}{p(\theta)}$
$= p(\theta/z)p(z)$

$p(\theta|z) = \frac{p(z|\theta)\pi(\theta)}{p(z)}$ (fixed → likelyhood, → PRIOR)

POSTERIOR

I can use this notation, play role of likelyhood $(p(z|\theta))$
$z \sim p(z|\theta)$

for many z:
$p(\theta|z_1 \cdots z_n) = \frac{p(z_1 \cdots z_n|\theta)\pi(\theta)}{p(z_1 \cdots z_n)}$

Date.

~~the~~
$z \in \mathbb{R}^d$
$\mathbb{R}^d \to$ ~~I~~ $\sim p(z|\theta_*)$, $\theta_* \sim \pi(\theta)$
$z_1 \cdots z_n \sim (p(z,\theta_*))^n$ i.i.d

derive new principle
idea: try to make prediction_ goal: posterior
$p(\theta|z_1 \cdots z_n) \propto$ (proportion) $\hat{L}(\theta) \cdot \pi(\theta)$
↓ exactly probability
$\prod_{i=1}^{n} p(z_i|\theta)\pi(\theta)$

~~wait~~
REGRESSION
$Y_i = N(\theta^T X_i, \sigma^2)$, $p(Y_i|\theta) = \frac{1}{c} e^{-(Y_i - \theta^T X_i)^2/2\sigma^2}$
$\Leftrightarrow (Y_i = \theta_*^T X_i + \varepsilon_i, \ \varepsilon_i \sim N(0,\sigma^2))$

$p(\theta|z_1 \cdots z_n) \propto \hat{L}(\theta) \cdot \pi(\theta)$
$= \frac{1}{c^n} e^{\frac{\sum_{i=1}^{n}(Y_i - \theta^T X_i)^2}{2\sigma^2}}$

how pick for θ. freedom, simple choice is Guassian (standard one)

$\pi(\theta) = \frac{1}{c'} e^{-\|\theta\|^2}$ take this and plug it here:

$p(\theta|z_1 \cdots z_n) \propto \hat{L}(\theta) \cdot \pi(\theta)$
$= \frac{1}{c^n} e^{\frac{\sum_{i=1}^{n}(Y_i - \theta^T X_i)^2}{2\sigma^2}} \cdot \frac{1}{c'} e^{-\|\theta\|^2}$
$= \frac{1}{c^n c'} e^{-\left(\sum_{i=1}^{n} \frac{(Y_i - \theta^T X_i)^2}{2\sigma^2} + \|\theta\|^2\right)}$

least square + regularization

Delijia

Title.

this is distribution, generate data,
try to find good estimator θ

~~maximize~~ MAP: MAXIMUM POSTERIORY

So: $= -\left(\sum_{i=1}^{n} \frac{(Y_i - \theta^T X_i)^2}{2\sigma^2} + \|\theta\|^2\right)$

$\max_\theta = \frac{1}{e^n C'} e$

Can be for logistic, ...

$Z_i = Z_m \sim (P_*)^n$

$P_* = P_{\theta_*} \quad \theta_* \in \mathbb{R}^p$

MLE $\max_\theta \hat{L}(\theta) = \prod_{i=1}^{n} P_\theta(Z_i)$

$Z = Y|X$

BAYESIAN $p(\theta|Z) \propto p(Z|\theta)\pi(\theta)$ (proportion)

MAP $\max_\theta p(\theta|Z)$

$Y \sim P_\theta(Y|X)$

REGRESSION

~~$P_\theta(Y|X) = \mathcal{N}(\theta_*^T X)$~~

$P_\theta(Y|X) = \mathcal{N}(\theta_*^T X, \sigma^2), \frac{1}{C} e^{-\sum_{i=1}^{n} \frac{(Y_i - \theta^T X_i)^2}{2\sigma^2}}$

Classification

$Y = (\pm 1) \quad P_\theta(Y|X) = \frac{1}{1+e^{-\theta^T X}}, \frac{1}{C'} e^{-\sum_{i=1}^{n} \log(1+e^{-\theta^T X_i Y_i})}$

, $P_\theta(\theta|Z_1 \ldots Z_n) = \frac{1}{C'} e^{-\sum \frac{(Y_i - \theta^T X)^2}{2\sigma^2} + \|\theta\|^2}$

noise fix the amount of Regularization

you could do variance formula.

there is another technique for Cross validation

MARGINALISED LIKLIHOOD, (in place of CV) cross validation

we dónot go to it.

story: ...

$(X_i, Y_i)_{i=1}^{n} \rightsquigarrow \hat{f}(X_{new}) \sim Y_{new}$

X-AI (explainable AI / interpretable / accountability)

SPARSity: bunch of zero
the target depends on few building blocks

Linear ~~LINEAR~~ models $f(x) = w^T x = \sum_{j=1}^{d} w^j x^j$

↳ coordinate of that perior

Take Vector Consider j from 1 to d (call it zero norm)

$\|w_*\| = \#\{j \in \{1-d\} \mid w^j \neq 0\} = \sum_{j=1}^{d} \mathbb{1}_{w^j \neq 0}(w)$

sparse vector — How many — (sum of steps) $-1 \leq x^j \leq 1$ $j=1$ $-d$

Delijia

Title.

$-1 \le x^j \le 1 \quad j = 1 - d$

$x^j \to E|x^j| = 0$

$E|x^j|^2 = 1$

SCALING MATTERS!

algorithm knows for sparcity

∧ for w that contains zero

$$\min_{w \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^{n} (y_i - w^T x_i)^2 + \lambda \|w\|_0$$

to calculate: take derivative equal to zero

the second part is sum of steps

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

$= \bar{w}^T \bar{x} \qquad \bar{w} = (w, b)$

$\bar{x} = (x, 1)$

Date.

Best subset selection (BSS)

Forturing the data until they confess

$J = \{1 \text{---} d\}$

$\hat{x}$, $\hat{x}^j$ j-th column of $\hat{x}$

$n \times d$

$\hat{y}_{n \times 1}$

BSS :

$\hat{J} = BSS(\hat{x}, \hat{y})$

$\hat{w} = \underbrace{LS}_{\text{least square}} (\hat{x}, \hat{y})$

$\hat{w}_i$ $i = 1 \dots 2^d$ → best error on training

↳ best possible error,

$n < d \to 0$

built model, check on error 30% of data validation.

we could implement it, but, you need subset function

$\{\hat{x}_{s_1} \cdots \hat{x}_{s_i}\} = \text{subset}(\hat{x}, s)$

$\{\hat{x}^1 \cdots \hat{x}^d\} = \text{subset}(\hat{x}, 1)$

solution: $\|\hat{y} - \nu \hat{x}^1\| \Rightarrow$

$n \times 1$ ν: number $n \times 1$

Title.

$error_j = \min_{v \in \mathbb{R}} \| \hat{Y} - v\hat{X}^j \|^2 \quad j = 1 \dots d$

just order them, and best simple 1

initialize $J_0 = \{0\}$ = empty set

$j_* = sort(err_j)$, update $J = J_0 \cup \{j_*\}$

$\Rightarrow$

$\hat{J} = BSS(\hat{X}, \hat{Y})$

$\{\hat{X}^1 \dots \hat{X}^d\} = subset(\hat{X}, J)$, $J_0 = \phi$

$error_j = \min_{v \in \mathbb{R}} \| \hat{Y} - v\hat{X}^j \|^2 \quad j = 1 \dots d$

$j_* = sort(err_j)$, $J_1 = J_0 \cup \{j_*\}$

$MP(\hat{X}, \hat{Y})$ MATCHING PURSUIT (GREEDY METHOD)

select 1 variable, add & update solution

$(\hat{w}) = MP(\hat{X}, \hat{Y})$

selected variable

Date.

$\hat{J} = \phi$ $\hat{r} = \hat{Y}$,

$\hat{w} = 0$; for $t = 1 \dots T$

$v_*^j = \arg\min_{v \in \mathbb{R}} \| \hat{Y} - v\hat{X}^j \|^2$ ← find best coefficient

$err_j = \min_{v \in \mathbb{R}} \| \hat{r} - v\hat{X}^j \|^2$; check error for each variable

$j_* = sort(err_j)$ → best column

I have to update j

pick column which is best

$\hat{J} = \hat{J} \cup \{j^*\}$, $\hat{r} = \hat{Y} - \hat{X}^{j_*} v_*^{j_*}$, $\hat{w} = \hat{w} + v_*^{j_*} e_{j_*}$

add column to select set update update in right place

this is one iteration

$T = t+1$ first

need to coefficient update

e_{j_*}

$\hat{w} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ \vdots \\ 0 \\ \vdots \end{pmatrix} + v_*^{j_*} \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 1 \\ 0 \\ \vdots \end{pmatrix} \leftarrow j_*$

orthogonal matching pursuit (recompute everything all the time)

$\hat{w} = \hat{w} + v_*^{j_*} * e_{j_*}$

$\hat{e} = \arg\min_{e \in \mathbb{R}^m} \| \hat{X}_{\hat{j}} a - \hat{r} \|^2$

$m = |\hat{j}|$

instead of comput seperate for every variable use all the selected one in 1 w ; put right number in right position

Delijia

Title. $\hat{r} = \hat{y} - \hat{x}_j \hat{\theta}$

if cost more - solve better

regularization parameter: $T >> d$, decide when to stop, trade of fitting data and sparsity

$$\min_{w \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^{n} (y_i - w^T x_i)^2 + \lambda \|w\|_0$$

$$\|w\|_1 = \sum_{j=1}^{d} |w_j|$$

LASSO

step size

① $$w_{t+1} = \underset{\lambda,\eta}{S} \left(w_t - \eta \nabla \| X w_t - \hat{y} \|^2 \right)$$

Gradient descent to pick s quietest

Soft thresholding of vector

entry small kill it

big: take alone, decrease abit

Small

λ: tells you how much threshold

λ: small & big ??

↳ more sparsity

itrative soft thresholding

② $$S_{\lambda\eta}(w)^j = \begin{cases} w^j - \lambda\eta & w^j > \lambda\eta \\ 0 & \|w^j\| < \lambda\eta \\ w^j + \lambda\eta & w^j < \lambda\eta \end{cases}$$

Review Sparse method

Date.

$$f_0(x) = w^T x \quad x \in \mathbb{R}^d$$

$$= \sum_{i=1}^{d} w^i x^i$$

x_1 = x position x_2 = area x_3 ... features

x_4 = How many cats #cats the ★ feature doesn not related to the question

$z_1 \dots z_n \sim \rho^m$

$$\hat{L}(w) = \sum_{i=1}^{m} (y_i - w^T x_i)^2 + \lambda \|w\|$$

Could be large small → pick 1, w → means

large → Consider all feature

Could be small

With λ you can have trade of between error & ??

$$\hat{L}(w) = \sum_{i=1}^{n} (y_i - w^T x_i)^2 + \lambda \|w\|$$

$$\lambda \sum_{i=1}^{d} |w_i|$$

LASSO

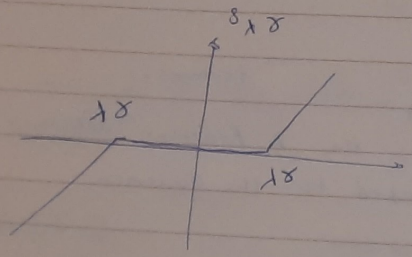
Delijia

Title.

ISTA: iterative soft thresholding Algorithm

$w_0 = 0$

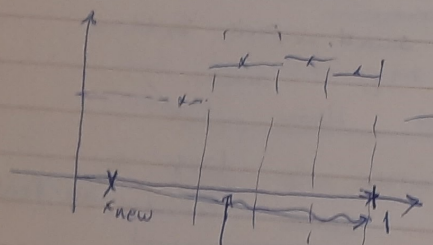
$$w_i = S_{\lambda\gamma}\left(w_{i-1} - \frac{2\gamma}{m} X^T (Y - W^T X)\right)$$



$$S_{\lambda\gamma}(x) = z \begin{cases} x - \lambda\gamma & x \geq \lambda\gamma \\ 0 & |x| < \lambda\gamma \\ x + \lambda\gamma & x \leq -\lambda\gamma \end{cases}$$

You only care about neighbours closs to him

local estimate algorithm



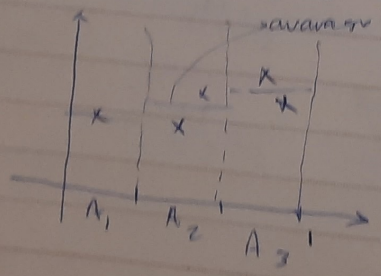
→ 4 cells to have partition

value of this period in on the line of this period

Def (Partition)

$$A = \{ A_j \; ; \; \bigcup_{j=1}^{J} A_j = X \; ; \; A_j \cap A_i = \phi \quad i \neq j \}$$

Given



after partition.

Date.

if we consider our estimator:

Piece wise constant function:

in partition we have different data

piece wise constant function

$$\min_{C} \sum_{x_i \in A_1} (y_i - c_1)^2$$

Consider data just in cell

$$\min_{c_1} \sum_{x_i \in A_1} (y_i - c_1)^2 \Rightarrow \hat{c}_1 = \frac{1}{n_{A_j}} \sum_{x_i \in A_j} y_i$$

how many points you have in A_1 → (~~how many partition you have~~)

$$f(x) = \sum_{j=1}^{J} c_j \mathbb{1}_{x \in A_j}(x)$$ → indicator function here

~~$x \in A_2$~~ $x \in A_2 \rightarrow f(x) = c_2$

J → how many partition you have

$$f(x) = \sum_{j=1}^{J} (w_j x) \mathbb{1}_{x \in A_j}(x)$$

piece wise linear function (or any function)

if partition is not good, find different partition?

you can design partition?

Delijia

Title: partition tree

$d=1 \quad x=[0,1]$

$A_1, A_2, A_3 \dots A_q$

A_q is a partition of x

$A \subset A_q$ there exist $c_1 \dots c_n \in A$ such that $A = \bigcup_{i=1}^{m} c_i$

$A_{1,1} = \bigcup_{i=1}^{2} A_{2,i}$

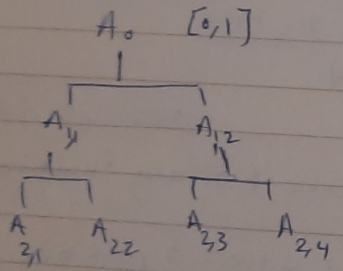
1/4 1/2 3/4 → uniform partition

$A_1 = \{ a = [0,1] \}$ is could be partition

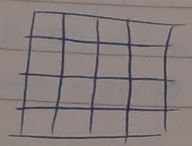
$A_1 = \{ A_{1,1} = [0, 1/2),\ A_{1,2} = [1/2, 1] \}$

$A_2 = \{ A_{2,1} = [0, 1/4),\ A_{2,2} = [1/4, 1/2),\ A_{2,3} = [1/2, 3/4),\ A_{2,4} = [3/4, 1] \}$

A we have depth, q



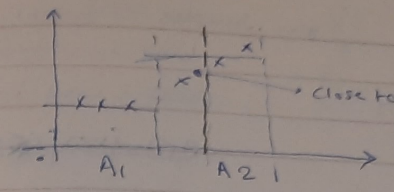
if Partion is big, its not good.



$q = 2$

Adaptive partition

Date:



close to the data below but can not jump above so we can not partition

I give you partition, compute loss piece wise function

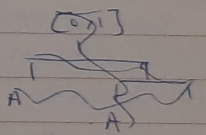
in this case can not partition it if we are in Uniform, because if we partition A_2, we must partition A_1

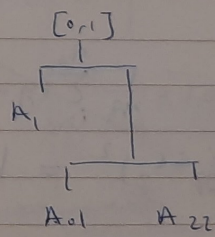
$$\text{Loss}_{A_1} = \frac{1}{n_{A_1}} \sum_{i=1}^{n_{A_1}} (y_i - c_1)^2 < \text{threshold}$$

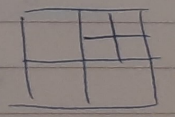
$$\text{Loss}_{A_2} = \frac{1}{n_{A_2}} \sum_{i=1}^{n_{A_2}} (y_i - c_2)^2 \geq \text{Threshold}$$

↓ do more partition

$A_2 = \{ A_{21}, A_{22} \}$



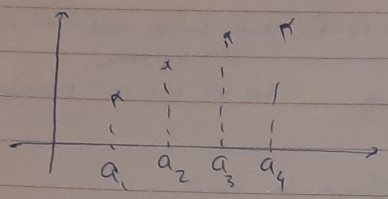




$q = 2$

Cut point $= \frac{t}{2^q}$ ← cut point

Title. Greedy approach: I give samples. I don't want to dispair, what I want to do



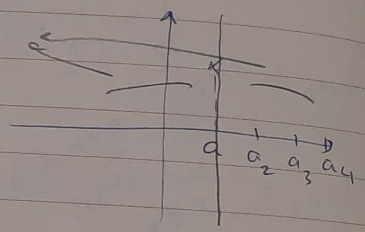
~~$A_1(s) = \{x \in X \mid x \leq s\}$~~

$A_1(s) = \{x \in X \mid x \leq s\}$, $A_2(s) = \{x \in X \mid x > s\}$

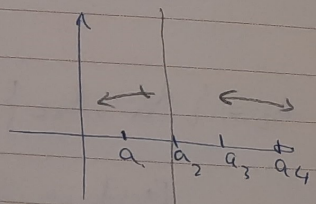
what I want to: choose cut point

$\min \hat{L}_{A_1(s)} + \hat{L}_{A_2(s)}$

$s \in \{a_1, a_2, a_3, a_4\}$



if cut from a_2



if $s = a_1 \Rightarrow \hat{L}_{A_1}(a_1) + \hat{L}_{A_2}(a_1) = \frac{1}{n_{A_1}} \sum_{i=1}^{n_{A_1}} (y_i - c_1)^2 + \frac{1}{n_{A_2}} \sum_{i=1}^{n_{A_2}} (y_i - c_2)^2 = L_1$

c_2: avarage value

a number $\Rightarrow$ avarage = 0 (phương sai)

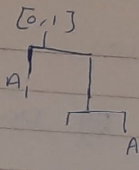
cut a_2

Compare L_1, L_2, L_3, L_4 → which is the smallest

cut a_3

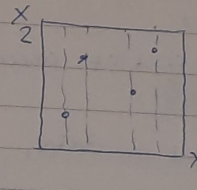
cut a_4

What tree took like



→ tree is not symetric

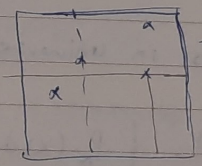
in dim = 2 →



If you have more dimention,

so minimizer not only depend on s, it depends on j

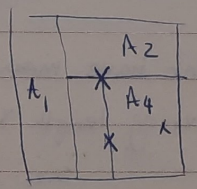
so (s^*, j^*) $\min_{s,j} \hat{L}_{A_1(s)} + \hat{L}_{A_2(s)}$



→ not good, loss is not good if we cut in this form

need compare loss & choose smallest, you can cut vertically or horizontally

maybe:



each cut make error smallest

Delijia

Title. DENSITY ESTIMATION:

I) $(X_i)_{i=1}^{n} \sim \underbrace{P_*}_{\text{unknown}}^{n}$ II) Density distribution part

Generative AI

chatgpt a kind of generative AI

find $\hat{P}$ based on data, P_*^n

find approximation $\hat{P}$

Cats I observed

II, $(\hat{X}_i)_{i=1}^{N} \sim \hat{P}^N$ SAMPLING part

$X = N$ $P_*(1) \dots P_*(N)$, $0 \leq P_x(i) \leq 1$; $\sum_{i=1}^{N} P_*(i) = 1$

all the possible cats in universe

$X \subseteq \mathbb{R}^d$ $A \subset X$ $P_*(A) = \int_A P_*(x)\,dx$ $P_x(x) \geq 0$

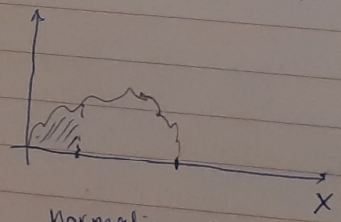
A: chunk of X

$\int_X P_*(x)\,dx = 1$

probibiliy, in each point dont have meaning must tak interval. , prob measuring length x value

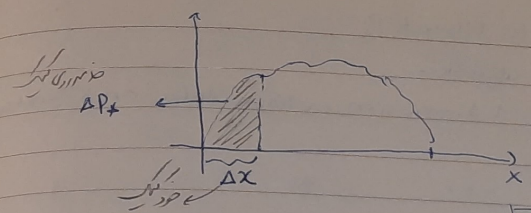
Size of point is zero

Size of Interval: may be $\frac{1}{4}, \frac{1}{3}, \dots$

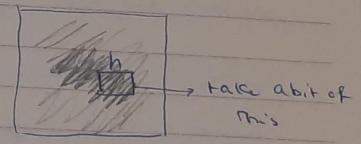


Normalize amount of stuff (all the parts) how much stuff is in (////) according to all the part

Date.



$\frac{\Delta P_*}{\Delta X}$



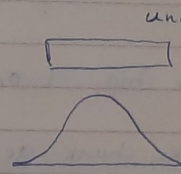
$\Delta x \sim h^d$

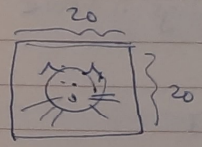
with out Creme, we have just cake and its Uniform. Normalize total amount.

Ex. $U([0,1]^d)$

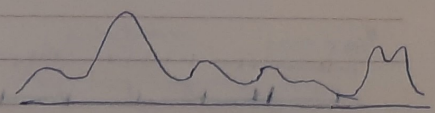
$e^{-\|x-\mu\|^2/2\sigma^2}$

lets draw in 1 dimension:





$d = 400$



multimodal

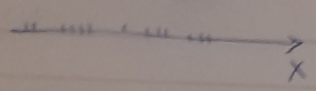
~~When~~ main goal maximum likelyhood

main reason that we haven't got label, an example of unsupervised learning.

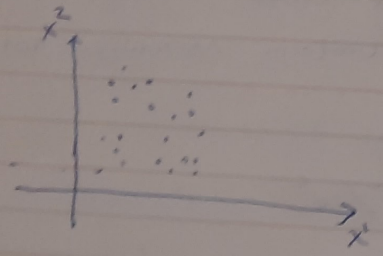
Title: lets invent way to estimate P_X

how could be invent algorithm.

lets draw it, in 1 dimension, we have data on X ...

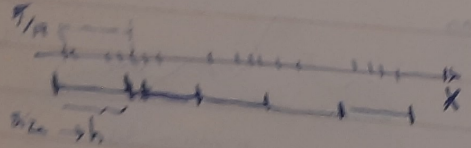


in 2 dim



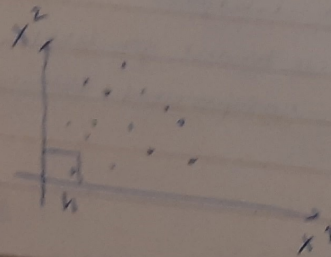
I give data, 1 or 1 dim;

take chunk, go in check how many things it have



take another interval: shift the first

(پنجره لغزان)



$$\hat{p}_h(x) = \sum_{j=1}^{N_h} \frac{n_j(h)}{h^d n} \mathbb{1}_{B_j(h)}(x)$$

number of point inside jth cell

$$n_j(h) = \sum_{i=1}^{n} \mathbb{1}_{B_j(h)}(X_i)$$

$$X = \bigcup_{j=1}^{N_n} B_j(h)$$

$B_i(h) \cap B_j(h) \neq \phi$ $i \neq j$ does not overlap.

Histograms

$$\begin{cases} X = \bigcup_{j=1}^{N_n} B_j(h) \\ B_1(h) \cap B_2(h) \dots B_j(h) \neq \phi \quad i \neq j \end{cases}$$

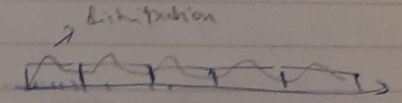
↳ Binning $B_j(h)$ bins

now select h : Cross validation.

there is a good h.

if $h \gg$, part: whole → Uniform distribution

in 1 dimension:



$\frac{1}{n}$ for each data: have a lot of spike.

Concern: partitioning & selection bandwidth (h)

dimension

Title. 3 10 >100

in dimension of 2, 3 this is ok

but think of d dimension must think different

d=10 : high for mathematic

very high dimension, completely of maybe nural network

~~off load~~ beat voting

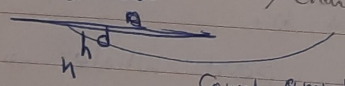
split domain, value? everybody vote.

Kernel Density estimator (KDE)

~~$\sum \mathbb{1}_{B_i(x)}(x)$~~ if I ask you how much is density is here based on data.

$$\hat{p}(x) = \sum_{i=1}^{n} \mathbb{1}_{\|x - x_i\| < \varepsilon} \quad \varepsilon < h$$

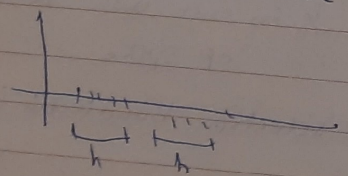
Change name.

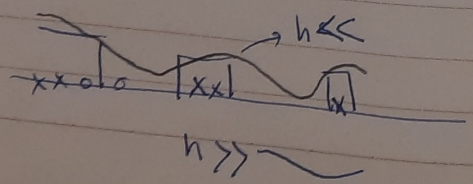


Count any thing $< \varepsilon$

divide by n

its like window, move





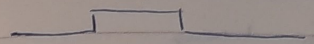
Date

$$\hat{p}_n(x) = \sum_{i=1}^{n} \mathbb{1}^{(x)}_{\|x - x_i\| / h \le 1} \rightsquigarrow$$

$$\hat{p}_h(x) = \frac{1}{n h^d} \sum_{i=1}^{n} K\left(\frac{\|x - x_i\|}{h}\right)$$

$K_h(\|x - x_i\|)$

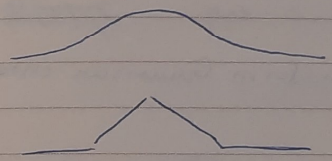
shapes:



~~$K(s) = \mathbb{1}_{|s| \le 1}(s)$~~

$$K(s) = \mathbb{1}^{(s)}_{|s| \le 1}$$

What are other shaps.



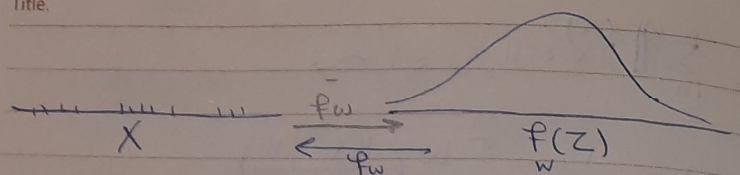
in all of this there are $\underline{h}$ which may stretch it.

every time give a number, compute distance to any, drow shap γ that is all.

~~Generative approach~~:

Title.

Generative approach:



Can I transfer Guassian into sth.

function of random variable

if random variable of Z in Gussion →

ran " " " X in linear

$$P_X = ? \; P_Z \qquad Z \sim N(0, I)$$

given density (P_z) go to another density P_x

f_i is invertable.

so I can go back. & transform Guassian into sth else.

measure distance between this two $(f_w(z), x)$

Why we call it Generative:

suppose I have f_w how to generate X

Sample Z, get f_w ang get X

this idea, Generative networks,

Normalizing Flow,

f_w is invertable so we have ← informula

if Z is Guassian, (Normal distribution

so we can have f_w

Generative idea.

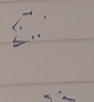
, normalizing flow

Date.

also called anomaly detection.

anomaly have unlabeld data.

ANOMALY Detection:



try to make algorithm

1; one of them is far away

2, if sparsely number, or high dimentia

find anomaly

How to find anomaly

~~$\hat{p}(x)$~~ put x, check $\hat{p}(x) \leq \tau$ then say Anomally

if big not anomally

$$a(x) = \mathbb{1}_{(\hat{p}(x) \leq \tau)}(x)$$

→ >> big anomally

How to choose τ.

Clustering: (grouping) group stuff

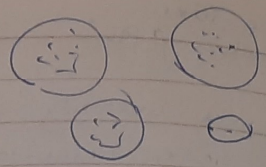
How could grouping, lot of algorithm

Find regions (distance small, Density high)

$$X_\tau = \{ x \in X \mid \hat{P}(x) > \tau \}$$

Delijia

Title: selected region



$I, X_\tau = \{x \in X \mid \hat{p}(x) > \tau\}$

II, find connected regions

DBSCAN

Unsupervised learning:

$\{x_i \rightarrow x_n\}$ Data, inputs, no output here

$\{x_i \rightarrow x_n\} \sim p_*^n$ Data come from Distribution.

$\{x_i \rightarrow x_n\} \sim p_*^n$

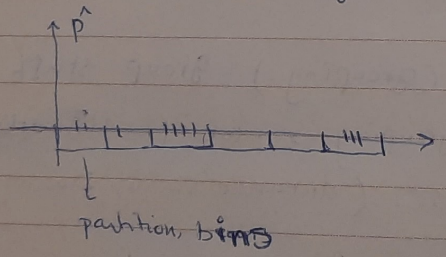
we would like to find $\hat{p}$ which is good appropriation of p_*^n

$\Rightarrow \{x_i \rightarrow x_n\} \sim p_*^n$

$\hat{p} \approx$

idea:

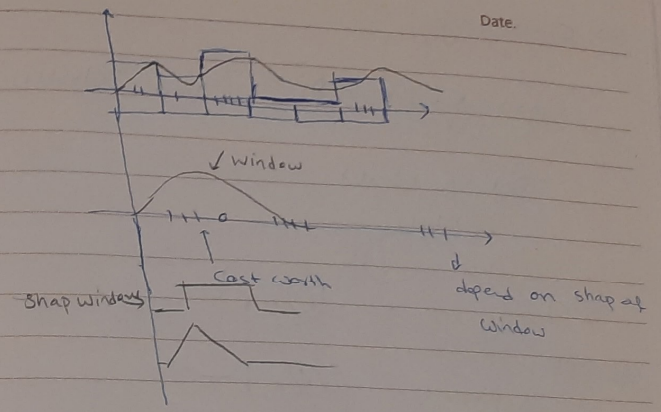
① partition Based → Histograms



Date:

② II

③ III



for 2: go inside each cell & count

Call Cell j ;

→ symbole: if it is there or not

$$M_j = \sum_{i=1}^{n} \mathbb{1}_{x_i \in B_j}$$

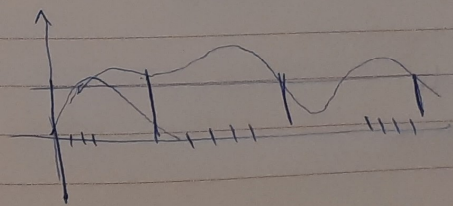
$\frac{M_j}{n} / \frac{1}{h^d} = \hat{p}_j$ → Density of each cell

for I

$$\hat{p}(x) = \sum_{j=1}^{\#Bins} \mathbb{1}_{x \in B_j} \hat{p}_j$$

for III

$$\hat{p}(x) = \sum_{i=1}^{n} \frac{K(\|x - x_i\|/h)}{n h^d}$$

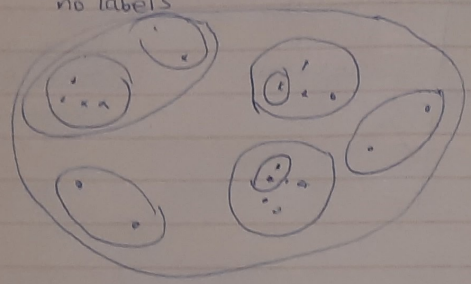


Title: Clustering:

I Given $x_i \rightarrow x_n \sim P_x^n$ find "Clusters"

Set of points

no labels

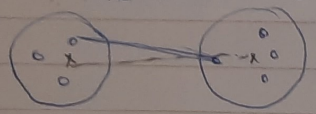


We can have different cluster, but do not when do statistic, its in sum kind its in some kind complicated.

give algorithm to group the points.

تا حد نقطه ها لازم هست بشود - کوچکترین فاصله اولین کلاستر ها که شود کلاستر بعدی

Hierarchical clustering تا حد کلاستر شود



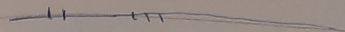
we call it: AGGLOMERATIVE CLUSTERING

between cluster: Linkage

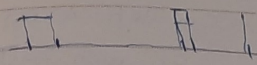
Linkage { AVERAG, SinGle, Complete }

DENDROGRAM

Date.



Five point in 1 dimension

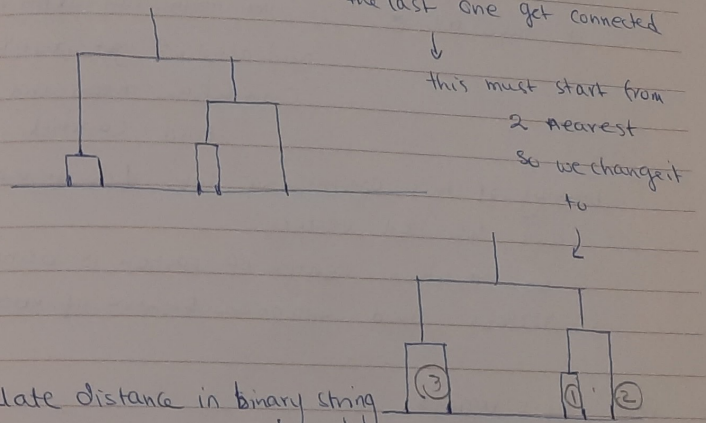


two points get connected

the second 2point get connected

the last one get connected

this must start from 2 nearest so we change it to



Calculate distance in binary string

$$\sum_{i=1}^{d} \mathbb{1}\, x_i^k \neq x_j^k = d_H(x_i, x_j) \qquad x_i, x_j \in \{0,1\}^A$$

second Algorithm:

K-means.

Statistical Algorithm ← idea

in Computation Call LLoYD's Algorithm ← Algorithm

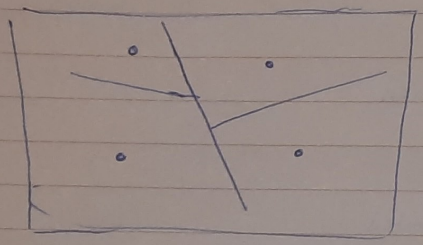
or Kmeans++

Algorithm

Delijia

Title. big challange: how could partition.

optimize possible partition.



we are in cube and there are partition we can defined capital of each partition

if take all : if get mean can defind partition and also if get partition can define mean and go to find capital

beauty of this optimizing, lets do gradients and we need vectors!

so partition, centre of pavtion is bunch of vectors and optimize centre of masses

Call V_j : regions

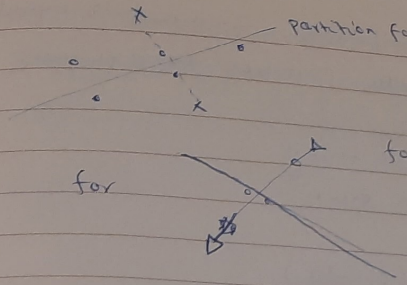
$j=1 \rightarrow K$ of them we need name for centers call them c_j

Which of this point is in clusters:

$$V_j = \{ x \in X \;\; \| x - c_j \| \leq \| x - c_i \| \;\; i \neq j \}$$
$$j = 1 - K$$

if I get you 2 partition, get different error

lets put some centers & give means. we can say pick 2.



put number.

I want function $\hat{L}(c_1, \ldots c_k) =$

take each point have to decide in which partition is this. So fix center, I :

$$\| x_i - c_j \|^2$$

among all centers, which is closest. so

$$\hat{L}(c_1, \ldots c_k) = \sum_{i=1}^{n} \min_{j=1-k} \| x_i - c_j \|^2$$

this is one way to do it.

this is very similar to loss function

$$\sum_{i=1}^{n} \min_{j=1-K} \| x_i - c_j \|^2$$

$$e(x_i, c_j)$$

Delijja

Title: what to do; minimize, gradient γ equal to zero

in infinit data, partitions are differents.

the most intuitive thing:

let's write down:

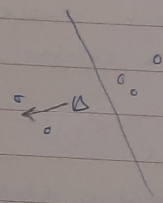
$$\min_{c_1 \dots c_k} \hat{L}(c_1 \dots c_k)$$

we have 2 minimization, one inside γ on outside

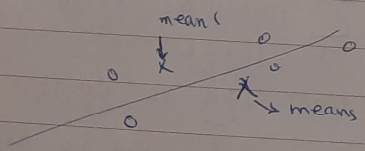
initialize the centers, compute the error

and update means.

find center of mass and keep on.



what you see get to right place and its easy



assign the means, reassign point → update means ↙ change the partition

init $c_1^0 \dots c_k^0$

assign $(x_1 \dots x_n)$ to $c_1^0 \dots c_k^0$ $\quad c_k^0 \mapsto \hat{V}_1^1 \dots \hat{V}_1^k$

update means, $(\hat{V}_1^1 \dots \hat{V}_1^k) \mapsto c_1^1 \dots c_k^1$

Date

so generalize:

- init $c_1^0 \dots c_k^0$

assign $(x_1 \dots x_n)$ to $c_1^{k-1} \dots c_k^{t-1}$ $\quad c_k^{t-1} \mapsto \hat{V}_1^t \dots \hat{V}_k^t$

update means, $(\hat{V}_1^t \dots \hat{V}_k^t) \mapsto c_1^t \dots c_k^t$

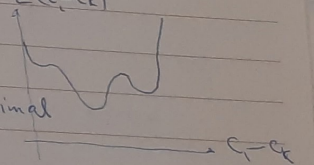
$$c_j^t = \sum_{x \in \hat{V}_j^t} x \cdot \frac{1}{\# \hat{V}_j^t}$$

what can you say about: $\hat{L}(c_1 \dots c_k)$

we

in situation

local minimal



its important for initialization

idea; choose initialization:

bad ; put all centers together is bad idea,

choose point from data,

and second is the furthers from it

third guy should be the furthers from 2 points

this algorithm is robust. by randomization

1 one pick at random, second one is pick of random.

Delijia

Title.

pick second one is farthers away.

if do avarage, shows that robust algorithm is easy but robust

I do:

$$c_1^0 = x_j \sim U[1 - n]$$

$$d_1 = \| x_i - c_1^0 \| \quad i \neq j$$

Probibility distribution:

so:

$$d_i = \| x_i - c_1^0 \| \quad i \neq j \rightarrow p_i = \frac{d_i}{\sum_{i \neq j} d_i}$$

Sample the next mean

$$c_2^0 \sim [p_1 \rightarrow p_{n-1}]$$

again compute.

K-means ~~is~~ initialization itself.

1, parameter tuning

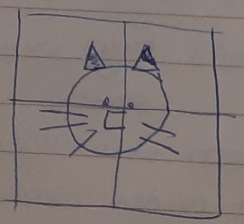
2, K-means

Suppose that image,

make image Compress

take pixel, clusters

10 clusters



for example the ears is in 1 cluster

the ones is black.

Very good in images.

its not a good examples of classyfying - for example cats & dogs

its good in 1 simple image for clustering

Date

Delijia