① Regression or Classification or Generative model

Regression: given $\bar{x}$ predict $\hat{y}$ — $\bar{x}$ = features, $\hat{y}$ = continuous

given weight & height → age

Classification: given $\bar{x}$ predict $\hat{y}$ — $\bar{x}$ = features, $\hat{y}$ = label +1/−1

given movie genre, movie actors } → yes/no like

② · Choose parameters → θ · the dimension of θ = the dimension of x (features).

determines the impt. of each features (weightage).

$$\boxed{\hat{y} = \vec{\theta}\cdot\vec{x} + \theta_0 = \underbrace{\vec{\theta}^{\,T}\vec{x}}_{\text{matrix mult}} + \underbrace{\theta_0}_{\text{bias}}}$$ general eqn form

$\hat{y} \rightarrow h(x, \theta)$

$\bar{x}$:

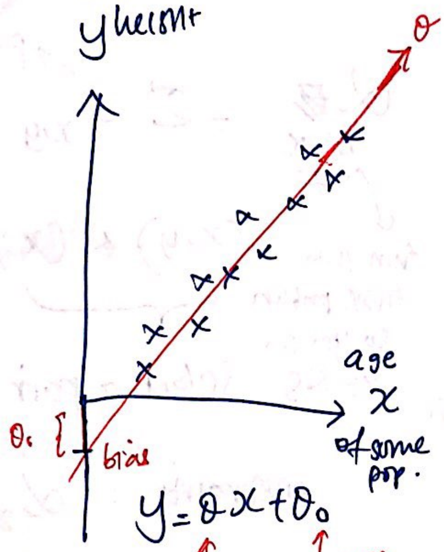
- genre · θ_1
- actors · θ_2
- time · θ_3
- country · θ_4
- theme · θ_5
- scriptwriter · θ_6
- director · θ_7

$+ \theta_0$

$\hat{y}$ = yes/no like. } C.

$\hat{y}$ = Rating } R

(R)



$y = \theta x + \theta_0$

↑ gradient ↑ y-axis int.

③ want to find out the $\boxed{\text{best}}$ θ

↳ how to grade what is good θ? or bad θ?

↓

$\boxed{\text{loss function}}$ ① $\min E_1 = \sum_{i=1}^{N} (y_i - \hat{y})^2 = \frac{1}{2N} \sum_{i=1}^{N} (y_i - (\vec{\theta}\cdot\vec{x}_i + \theta_0))^2$

y_i: true value of y from train set for train loss

: min least sq.

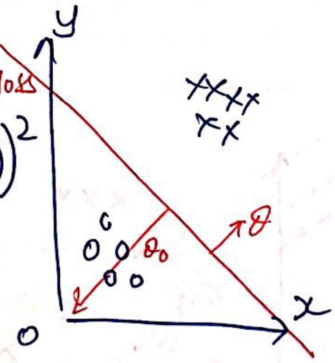
For training loss, use training set.

— test loss use test set.

(unbounded) linear regression.

(x_i, y_i) → data given.

(C)



② $\min E_2 = \sum_{i=1}^{N} \max \{1 - (\underset{\text{true y}}{y}(\underbrace{\theta\cdot x}_{\text{predicted y}})), 0\}$

perceptron ← (−1/+1)

a good θ will give $\theta \cdot x$ as the same sign as y

③ $\min E_3 = -\frac{1}{n}\sum_{i=1}^{N} \log\left[\frac{1}{1+e^{-y(\theta \cdot x)}}\right]$ } $\underbrace{\text{logistic}}_{}$ regression.

bounded btw 0 to 1

$\underbrace{}_{\text{Rating}}$.

$\frac{dE_3}{d\theta}$ ... ?

(better).
Classifier

④ $\min E_4 \Rightarrow$ SVM, parameter is called α.

dimension of x < $\alpha\theta$s.

max margin because margin $= \frac{1}{\|\theta\|}$

error fn $\Rightarrow \min \boxed{\frac{1}{2}\|\theta\|^2} = \min E_4$. / constraint $y(\theta^T x) \geq 1$.

↓ transform to max problem. ~~cannot GD~~.

$$\max \alpha = \sum \alpha_{x,y} - \frac{1}{2}\sum_{x,y}\sum_{x'y'} \alpha_{xy}\,\alpha_{x'y'}\, y\, y' \underbrace{(x^T x')}_{\text{a number}}.$$

params params

turn into max problem so you can use θs

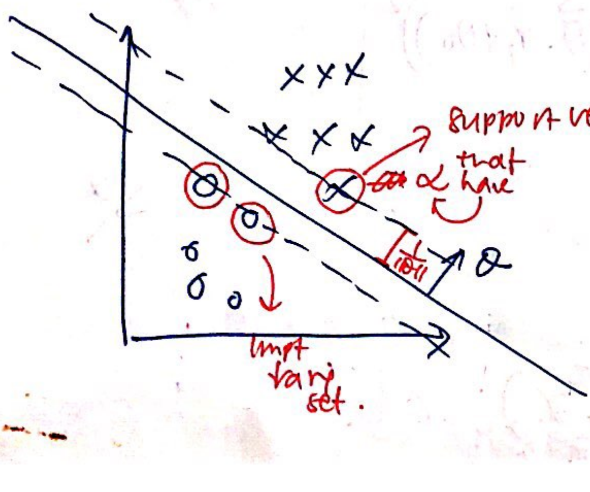
$\underbrace{(x,y)\ \&\ (x',y')}_{\text{select a pair}} \rightarrow$ training set

$$[x_1 \ldots x_d]\begin{bmatrix} x_1 \\ \vdots \\ x_d \end{bmatrix}.$$

constraint : $\alpha_{x,y} \geq 0$.

If I have N items in training set, I have N αs.

$$x = \begin{bmatrix} x_1 \\ \vdots \\ x_d \end{bmatrix}$$



test / train features

$$X = \underbrace{\begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1d} \\ x_{21} & \cdots & \cdots & x_{2d} \\ & & & \\ x_{N1} & \cdots & \cdots & x_{Nd} \end{bmatrix}}_{}$$

features

# of train/test data

$$Y = \begin{bmatrix} -1 \\ 1 \\ \vdots \end{bmatrix} \Big\} N \qquad \theta = \begin{bmatrix} \theta_1 \\ \vdots \\ \theta_d \end{bmatrix}$$

④ · After getting loss fn : want to min loss.
↓
differentiate the loss fn.

~~↳ ① Exact soln.~~

~~Step ①~~ $\frac{dE}{d\theta}$ and $\frac{dE}{d\theta_0}$ } find.

$\frac{dE}{d\theta_1}\left(y_i - \underbrace{\theta_1 x_1 + \theta_2 x_2 + \theta_0}\right)^2$

$\frac{dE}{d\theta_1} = \downarrow 2x_1$

the error of all training set. ← $\boxed{E_1} = \sum_{i=1}^{N} \left(y_i - (\theta \cdot x_i + \theta_0)\right)^2$

the error of one θ ← $\frac{dE_1}{d\theta} = \sum_{i=1}^{N} 2\left(y_i - (\theta \cdot x_i + \theta_0)\right) \cdot -x_i$

⋮

do the same for ALL parameters.

method ① Exact soln : equate $\frac{dE}{d\theta}$s $= 0$ and find θ_s.

method ② gradient descent :

initialize all θ_s params to some value. (any value)

repeat for K iterations or until converge

Check current Error → sub θ to $E(\theta, x, y)$ → sum all errors from TS.
← training sets (ALL)

if $E > e$ → some small value → 0.0001
(if still have error)

↳ update θ

$$\theta^{new} = \theta^{old} - \eta \boxed{\frac{dE}{d\theta}}$$

↙ sub in θ old values and sub in (x, y) from training set.

minus of the error of this θ from the "old" θ value to make up the new θ value.

constant, given, e.g: 0.01 or 0.05 . . .

else, (if no more error).

end.

⑤ Backprop (NN).

③ Stochastic GD → sub θ to $E(\theta, x_i, y_i)$ → choose 1 point, (Random) no sum

Fast ④ Quadratic solve of have constraint.

Conv. feature trick → to code nicer :

$$\vec{\theta} \cdot {}^{i}\vec{x} + \theta_0 = y$$

$\downarrow$

ith data

$$\begin{bmatrix} \theta_1 \\ \vdots \\ \theta_d \end{bmatrix} \cdot \underbrace{\begin{bmatrix} {}^{i}x_1 \\ \vdots \\ {}^{i}x_d \end{bmatrix}}_{\text{dimension}} + \theta_0 = \hat{y}$$

fold

$$\underbrace{\overset{\vec{\theta}}{\begin{bmatrix} \theta_1 \\ \vdots \\ \theta_d \\ \theta_0 \end{bmatrix}} \cdot \overset{\vec{x}}{\begin{bmatrix} x_1 \\ \vdots \\ x_d \\ 1 \end{bmatrix}}} = \hat{y}$$

← append.

nicer to code.

$$X = \begin{bmatrix} x_{11} & \cdots & x_{1d} & 1 \\ \vdots & & & \vdots \\ x_{N1} & \cdots & x_{Nd} & 1 \end{bmatrix}$$

np.dot(θ, x)

$$\vec{\theta} = \begin{bmatrix} \theta_1 \\ \vdots \\ \theta_d \\ \theta_0 \end{bmatrix}$$ → bias.

$$\frac{dE_2}{d\theta} \begin{cases} 0 & \text{if } y(\theta \cdot x) > 1 \\ -yx & \text{if } \underbrace{y(\theta \cdot x) < 1}_{\text{wrong prediction}} \end{cases}$$

correct prediction.

in GD, update θ:

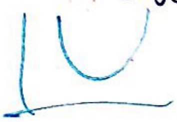
$$\theta^{new} \leftarrow \theta^{old} - \eta \boxed{\frac{dE}{d\theta}}$$

→ will be 0 if $y(\theta^{old} \cdot x) > 1$

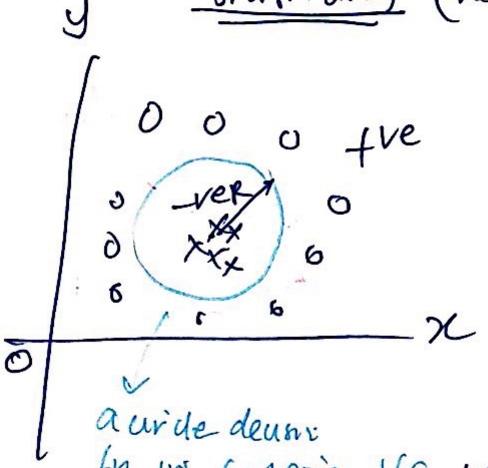
---

kernel → transform from ~~one dim~~ one vector space to another vector space where in that new vector space, the data is linearly separable.

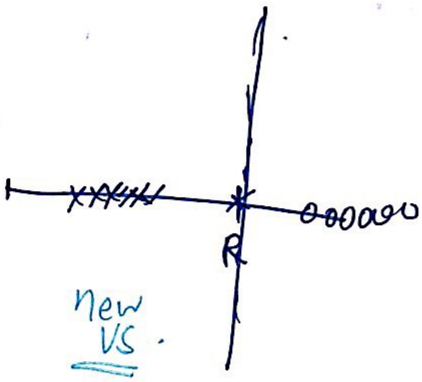
↳ gets the $x^T \cdot x$ in the new VS



straight away (cheat).



a circle drawn by us in Cartesian VS ~~kernel~~.



new VS.

transformation..

$x^2 + y^2 = R^2$

$(x_1, y_1) \rightarrow x_1^2 + y_1^2 = R_1^2$ my new data.

(x_2, y_2)

$\vdots$

$(x_N, y_N) \rightarrow x_N^2 + y_N^2 = R_N^2$

label is +ve -ve same value.