Saleprice = Qty * Saleamt (1 - D) , Costpr. = Qty * Costamt , % t = Sal A + Cost A

M - Sale A - ca

Bharat Gandhi

1) Regression project : Not time series data, y is cont.

2) $a^0 = 1$

$0^b = 0$

3) $0^0 = ?? = 1$

$0.9^{0.9} = 0.909$ $0.6^{0.6} = 0.736$ $(0.2)^{0.2} = 0.72$

$0.8^{0.8} = 0.836$ $0.5^{0.5} = 0.707$ $(0.1)^{0.1} = 0.79$

$0.7^{0.7} = 0.779$ $(0.4)^{0.4} = 0.6963$ $(0.01)^{0.01} = 0.95$

$(0.3)^{0.3} = 0.693$ $(0.001)^{0.001} = 0.993$

4) We cannot compute this, we have to use limit

$\lim_{x \to 0} x^x = 1$, as x tends to 0, it approaches to 1.

$\lim_{a \to 0} \frac{1}{a} = \infty$

Derivatives

5) Derivatives

$\frac{d}{dx} f(x)$

Rate of change

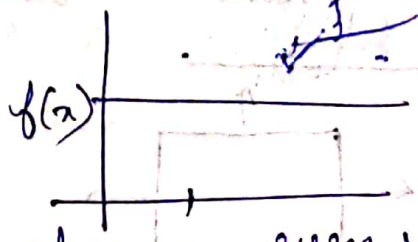
infinitesimal change in x

What is the change in f(x) when x changes by infinitely small amount

or infinitesimal

[∴ infinitesimal is $\frac{1}{\infty} \to 0$]

6) $\frac{d}{dx} c = 0$

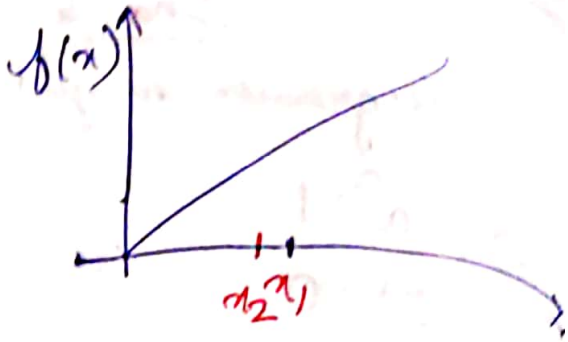


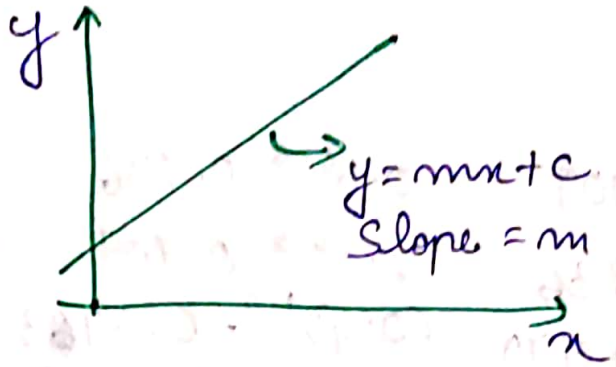
f(x) doesn't change even if we change x by small amount.

1) $\frac{d}{dx} f(x) = \lim_{\Delta x \to 0} \frac{f(x+\Delta x) - f(x)}{\Delta x}$

2) $\frac{d}{dx}(x) = 1$

$\frac{x_2 - x_1}{x_2 - x_1} = 1$

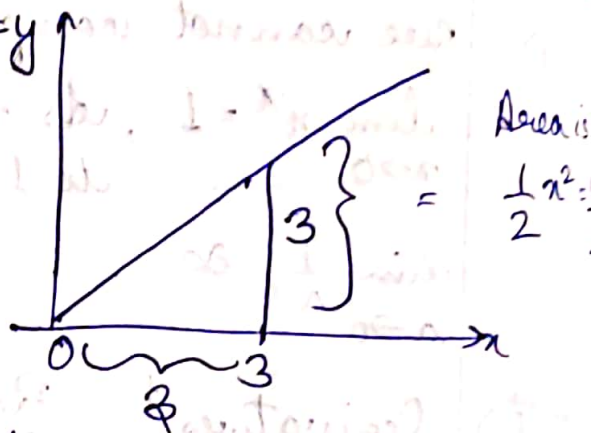




$\frac{dy}{dx} = m$ (Slope)

Slope of curve is the rate of change

INTEGRALS: $\int f(x)dx$



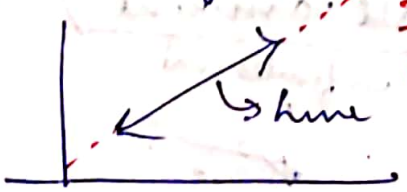
③ $\int x\,dx = \frac{x^2}{2}$

④ $\int_0^3 x\,dx = \frac{9}{2}$

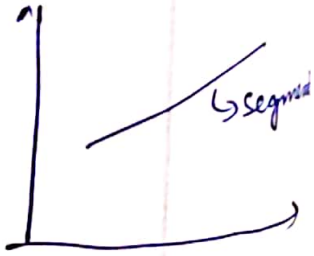
Imp ⑤ Integrals tells us the area under the curve

LINES & PLANES

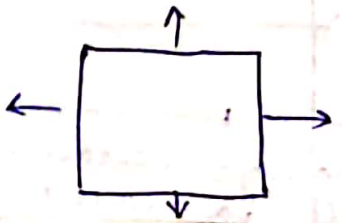
A line can expand either of the two direction



It cannot move in an orthogonal direction

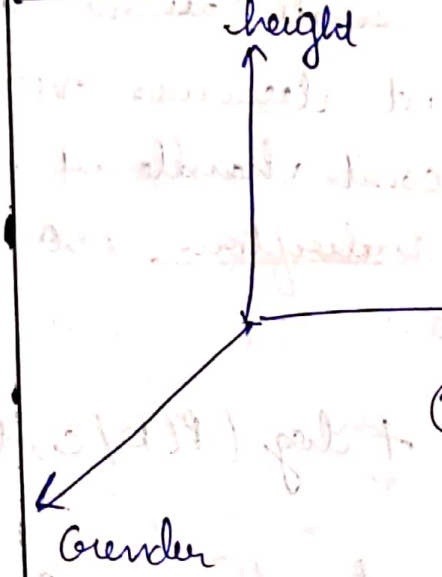


PLANE:



It cannot move in an orthogonal direction

① An n-dimensional plane can move only in those n-dimensions, not in other direction
Ex: 5-d can move only in 5 directions.

② 

① It should learn $z=0$ is a boundary
② Classification would only be possible on the orthogonal sides
③ On one side of orthogonal sides one class & other side of " " is another class.

LOGARITHMS

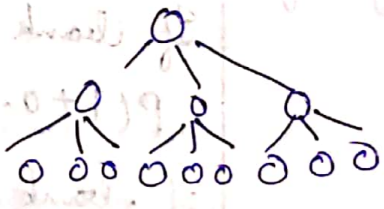
$\log_{10} 100 = 2$ (Representing 100 in decimal it takes 3 digits $(2 + \frac{1}{\text{always}})$

$\log_2 1000 = 9 - 9$. (representing 1000 in binary)

Proper def:: ← How much does it takes to represent certain amount of information

$\log_3 9 = 2$

$\log_3 81 = 4$
↳ number of leaf nodes



(Becoz this is a ternary tree, that is it splits into 3 always)

Properties of log.

1) A very large number, if we represent in log is very small, a very small nos. represented by log makes it large. either way is true that is

$\log_{10} 0.\underbrace{\ldots\ldots}_{\text{0 followed by 100 zeroes}} 1 = -100$

Chain Rule of Probability

$P(a, b, c, d, e) = P(a/b, c, d, e) \cdot P(b/c, d, e) \cdot P(c/d, e) \cdot P(d/e) \cdot P(e)$

$P(a/b, c, d, e)$ = is not possible since the prob. are very small (assume 10^{-2}) and multiplied 4 times, it becomes very small. So, our computer cannot handle it. So, computer encounters underflow, so the o/p will be always zero.

So to solve this problem:

$\log(P(a/b, c, d, e)) + \log(P(b/c, d, e) + \ldots$

① So, log comes in handy in case of NLP task.

Euler's No. e

$a^1 = a$

(Transcendental Number)

Origin of e

$P(1+r)^n$

If bank gives 5% rate of interest for 2 years

$P(1+0.05)^2$

If bank gives once a quarter in these 2 years

$P\left(1+\frac{0.05}{4}\right)^8$

If we say give every month in these 2 years

$P\left(1+\frac{0.05}{12}\right)^{24}$

On as we increase this like every hour, minute & every second. So, we keep dividing the base & multiply the power,

① At a certain point it will not increasing & stops at a particular number that is 'e'

Vectors and Matrices

① Vectors : which has magnitude & direction

② $x \quad y(x)$

① Matrices are functions for vectors, which will transform these vectors.

② They will take a vector & then transform into a vector (for ex, it gives the importance of the features eigen value)

---

linear

Eq^n of line:

$y = mx + c$

① Linear is not necessary to have everything to do with the line.

Eq^n of plane:

Here, 3-dimensional

$y = a_1x_1 + a_2x_2 + a_3x_3 + a_0$

linear doesn't do with line, it basically deals with power (that is always 1)

*** linear will not create a curve, we can only have line & plane

---

Arith matic Mean $\frac{a+b}{2}$

Harmonic Mean $\frac{2}{\frac{1}{a}+\frac{1}{b}} = \frac{2ab}{a+b}$

When to use what

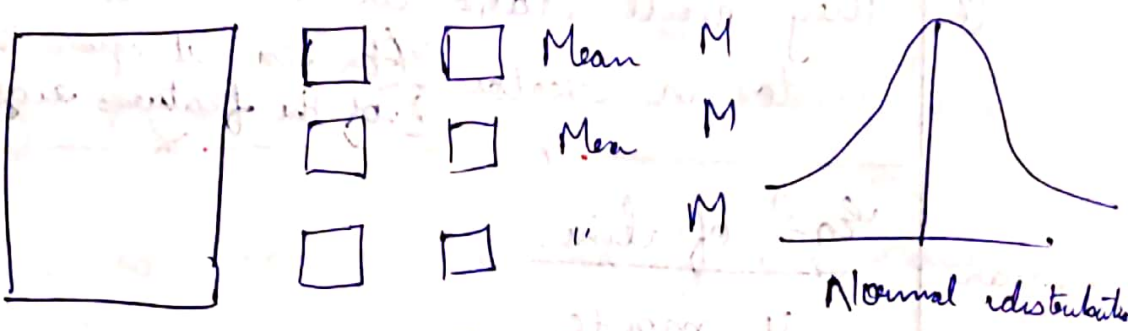
$\frac{\text{dist (Const.)}}{\text{time (A variable)}} = \text{Speed}$

Properties of H.M

If we have bunch of positive numbers, Harmonic mean is smaller than A.M. It tends to smaller values. It doesn't gets effected by outliers as much as Arithmetic mean does. (In case of outliers H.M is used) → only in ~~x~~ positive outliers.

Here, Median is expensive so we will use H.M.

Central Limit Theorem



Std. Error

$\frac{\sigma}{\sqrt{n}}$, Suppose, $n = 100$

so, $\frac{\sigma}{\sqrt{n}} = 0.1\,\sigma$

if $\sigma \downarrow$, then $0.1\sigma \downarrow\downarrow$

So, we want $\sigma \uparrow$, then 0.1 is significant

↓ original dist is very spread out.

↓ so we might miss out some region if we don't use divided by $\sqrt{n}$.

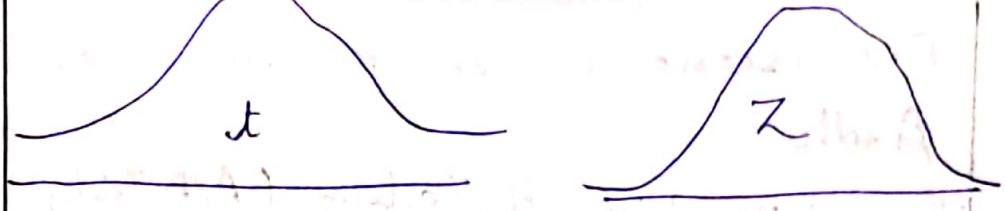
Z-test t-test

t → slightly more spread out (student t-dist)

Z → less spread out.

If small sample size is there, we want to have slightly more spread out dist, so as to capture all things in that small size sample data. (Moreover when we don't known Std. Dev. of population)

30 sample size !?



tails are broader — tails are almost touch to the axis.

Answer: → They could fit 30 values in a paper;
They were fitting & creating t-table & z-table
They didn't use another paper

So here cutoff was arbitrary

---

so, anywhere between 20 to 40 can have z test
But, in case of very less value z-test will perform worst.

Hypothesis Testing

Null

Alternative

p-value: Given the results how likely is it that the null hypothesis is not true

If prob (< 5% — Arbitary std. number used) not proved

It depends (some company uses 1%, " " " 10%.

A/B Testing (Hypothesis is used most)

1) For instance, we will say we want to change the button of buy now in Amazon.
Some pop. see it in below
" " " " " above
Then see the order &
see the conversion

One scenario conversion was 0.2
Another " " " 0.2001
Then we use H. Testing. (A/B Testing)
What is the prob. of value 0.2001 coming

Here sample size should be sufficient.
Demography is not considered. ✗

Linear Regression

Assumption : No Autocorrelation
But in Time series data we must have autocorrelation, so we cannot use Lin. Reg. on Time series data.

Homoskedasticity : Variance of a variable should be constant

** The residuals need to be normally dist.

Why ✓ What model would learn from training data it would not perform well on testing data.

Hypothesis $h(x) = \theta^T x$ — Cost function

Minimize Error

$$J(\theta) = \frac{1}{2m} \sum_{i=1}^{m} (y_i - \theta^T x_i)^2$$

Why Square → 4 will give much more minima than square, so we will use only 2.

$\sqrt{25} = \sqrt{-5}\ \sqrt{-5} = i\sqrt{5} \times i\sqrt{5} = -1 \times 5 = -5$

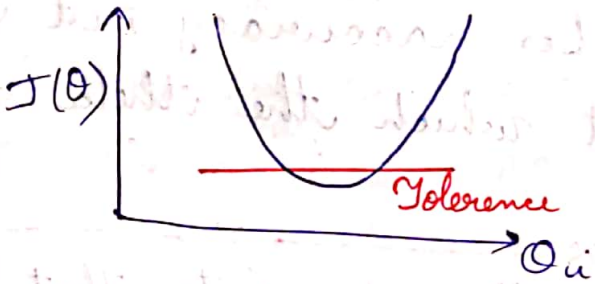
Gradient Descent

$$\theta_i := \theta_i - \alpha \frac{\partial J(\theta)}{\partial \theta_i}$$

α → Learning Rate

$\frac{\partial J(\theta)}{\partial \theta_i}$ → Gradient / slope = Direction & Magnitude

$\frac{d}{dx}$ = (derivative) $\quad \frac{\partial}{\partial x}$ (Partial Derivative) when we have multiple values of x



In curve, it takes very close data points which is basically a line & we calculate the slope.

Why learning Rate?

1. We can control of how much change we want.
2. Without α also it will work.

Where to stop?

1. When $\theta_i := \theta_i$ are approximately close to each other we will stop. (That is Tolerence)

There are 2 variable

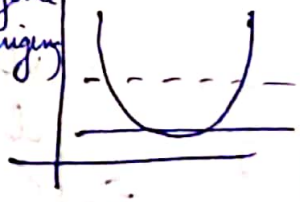
1. Learning Rate
2. Tolerance = If change bet^{n} theta is less than one particular value, i.e. 0.001 then we will stop

---

If α is low & high value of tolerence

i) Accuracy will be low.
ii) Convergence time will be more.

α is high, Tolerance is low
i) Convergence is challenging
ii) Acc. is fine

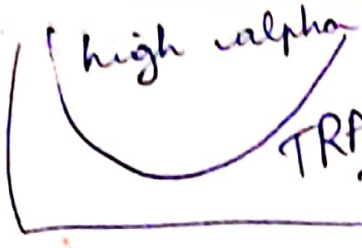


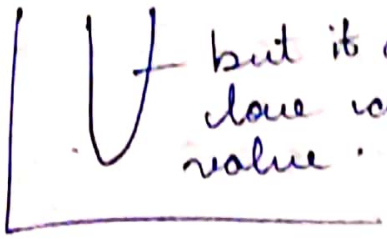
If α is high tolerence is also high

i) It will converge very quickly.
ii) Accuracy will be low

α is low, Tolerence is low
i) Extremely more time to convergence
ii) Acc. is reasonable

Scaling





1. Same alpha value cannot work for both the features of different scales
2. So, this is the reason we scale the features.
3. α don't describes accuracy, it describes the speed at which the theta will converge.

Overfitting

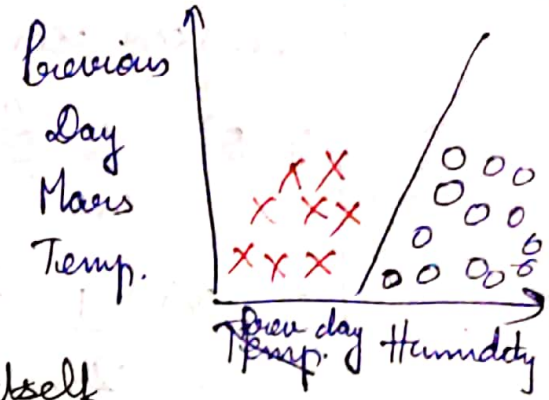
Noise: Is basically the data that is unexplainable by given variable. → the variable which is unable to explain the data.

Source of Noise:

1. Data input / tracking error
2. Calibration Error [Ex: Temp. calibration]
3. Survey error [Ex: Smoking]
4. Unaccounted variables [Ex: lung Cancer prediction then a person who hasn't taken cigar, smoke, drink also had shown lung Cancer, because to the asbestos factory to which he was living. That is for our model it is noise]
5. Since, we are telling that we are giving all variable, it (model) tries to fit everything. Thats why model has noise. We said that our objective is to fit everything so that training error is less.

What stops the model from overfitting

① Nos. of features ↓
② Degree of polynomial ↓
③ Complexity of the Model ↓



Complexity ↓
The amt. of the data model needs to store to save itself

2 variables, Degree = 2

6 Theta values $\{\theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_2^2 + \theta_5 x_1 x_2 + \theta_6$

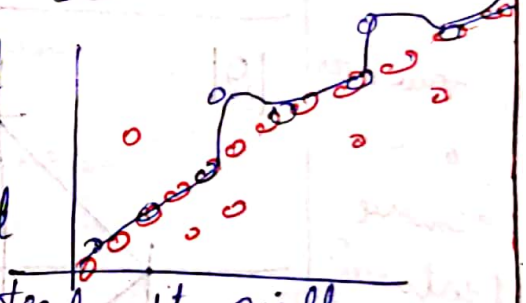
Avoiding ① Increasing the size of the training data.

Justify of Above Statement:

Charac. of Noise: Random (It will look Random to the model)

The model is unable to spot these noise patterns, that's why noise is random here.

(Ex: Generating pattern for 3 random nos. is easy than " " " 30 " numbers)



On inc. the data pts, model will find difficulty to fit the noise, so model will not go to fit the noise instead it will fit the signal those are many in nos; model cannot ignore that signal

i) ① Reduce Complexity ↓ (underfit)
② " Features ↓
③ " Degrees ↓
④ " Hidden Nodes
⑤ " Tree Depth

For underfitting only (pdf(i)) the other way works point (ii) & (iii) won't work

ii) Inc. training data

iii) update objective function (Regularization: don't just reduce error

instead reduce error plus something.

Regularization

$$J(\theta) = \frac{1}{2m} \sum_{i=1}^{m} (h(x_i) - y_i)^2 + \lambda \sum |\theta|$$

to differentiate this it have to do conjugate derivative since its not a continuous func

$$J(\theta) = \frac{1}{2m} \sum_{i=1}^{m} (h(x_i) - y_i)^2 + \lambda \sum \theta^2$$

to differentiate this it is easier since its a continuous func

Lasso Vs Ridge

~~Ridge:~~ When there is multicollinearity & both the features are important then use ridge.

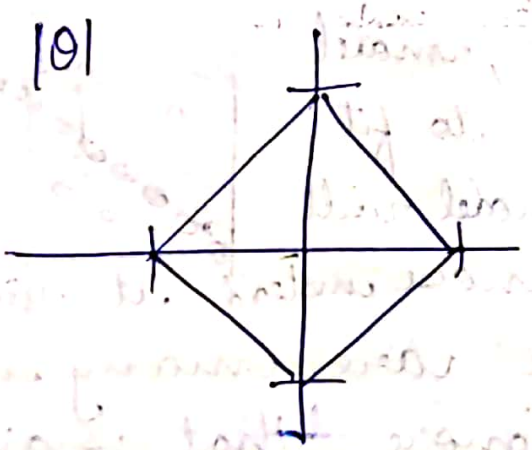
② Speeder

Lasso: ① We can use lasso for feature selection

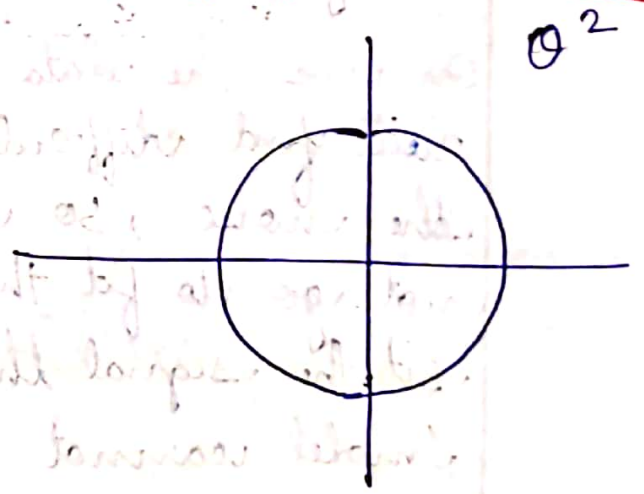
② Nos. of features > Nos. of obs.

③ Sparse: When there is data points that is widely spread in multidimensional with lot of zero element.

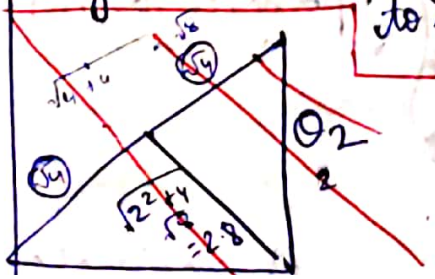
How does lasso Remove Features?

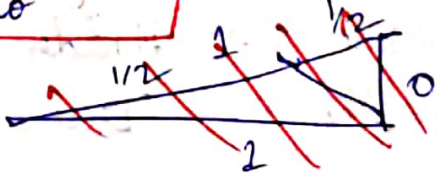


Here minima has only 4 maxima point. It can pick any value out of this 4 points to make variable zero



Here there is no minima





Ridge Vs Elastic Net

Ridge	Elastic
① less tuning	① Better in case of sparse dataset.
② Speed	

→ You reduce the dimension its more denser. } We reduce the dim. So that we make more dif. for model to fit it.
You inc. the " " " sparser

→ So kernal func. inc. the dim., so easier to seperate data.

Vectorization

① I have to compute price of all used cars.

Next Class:-

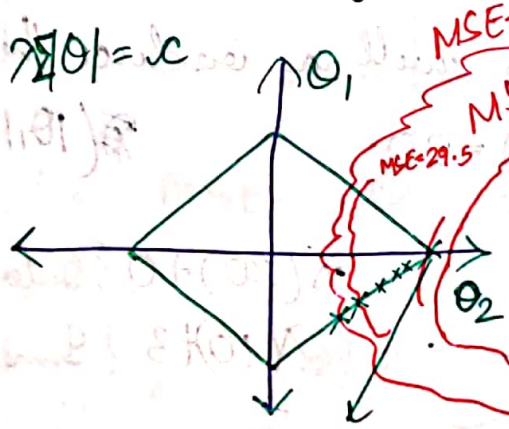
kasso, Ridge, AIC, bic, onehot vs labelEnc, Elastic net, logistic DTree

---

12/08/19

kasso & Ridge Clarification

$\lambda|\theta| = c$



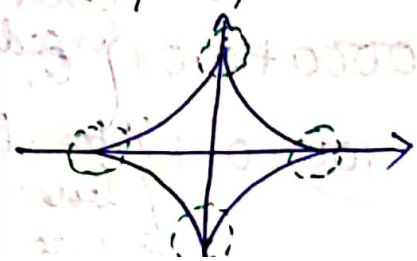
$J(\theta) = \frac{1}{2} MSE + \lambda\theta^2$ → 'Ridge'

$J(\theta) = \frac{1}{2} MSE + \lambda|\theta|$ → 'kasso'

Finally we will stop at this point.

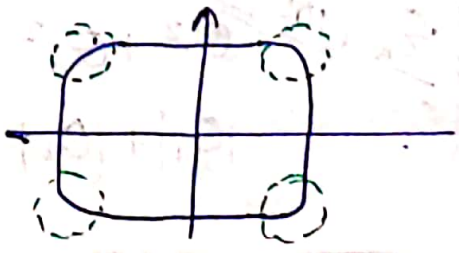
$L_{0.5}$

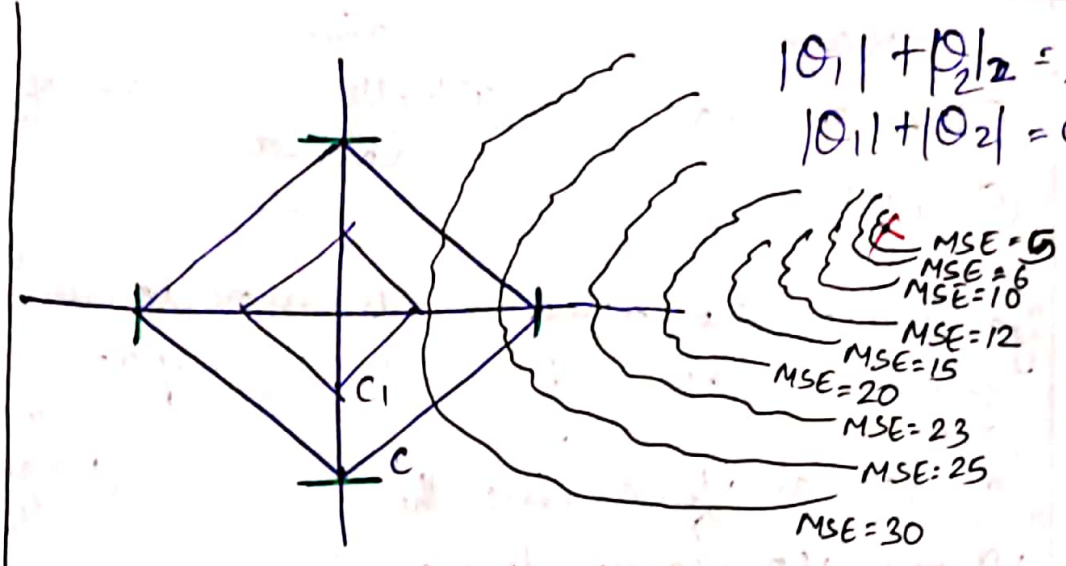
$\lambda\Sigma|\sqrt{\theta}|$



L_4

$\lambda\Sigma\theta^4$



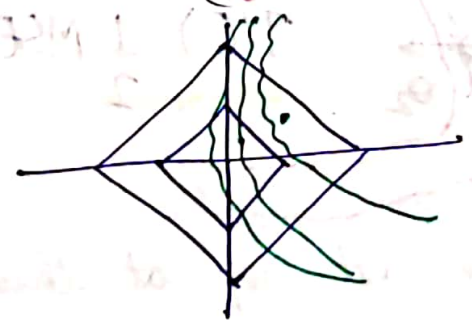


X : If I dey reach this point then we will end up increasing θ_1 and θ_2

→ So, we dey try to keep the contour within the diamond so that it is at the minimum end of the diamond

→ Now, there is reg. so we can increase MSE so that it intersect the diamond

→ In reduce MSE, as well as reduce the sum of thetas Ex, $\hat{\theta}(|\theta_1| + |\theta_2|) = 20$ $\hat{\theta}(|\theta_1| + |\theta_2|) = 10$



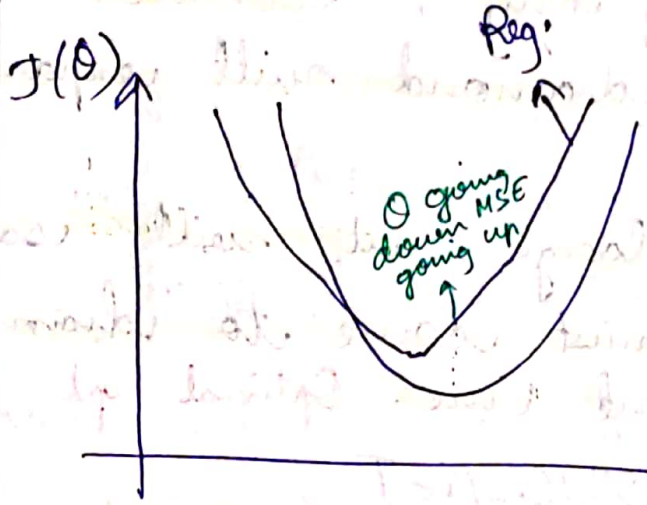
$\lambda(20) + 0$ {Outer diamond}
$\lambda(10) + 3$ {Inner diamond}

→ $\lambda 0.5$ = It is more stricter, so it will turn θ to zero more often than lasso

Ex. $\frac{100}{=}$ $\frac{0.1}{=}$ $\theta_1^2 + \theta_2^2$ $10000 + 0.01$ {For ridge, it needs to reduce θ_1

$|\theta_1| + |\theta_2|$ $100 + 0.1$ {For lasso, it don't bother much either to reduce θ_1 or θ_2

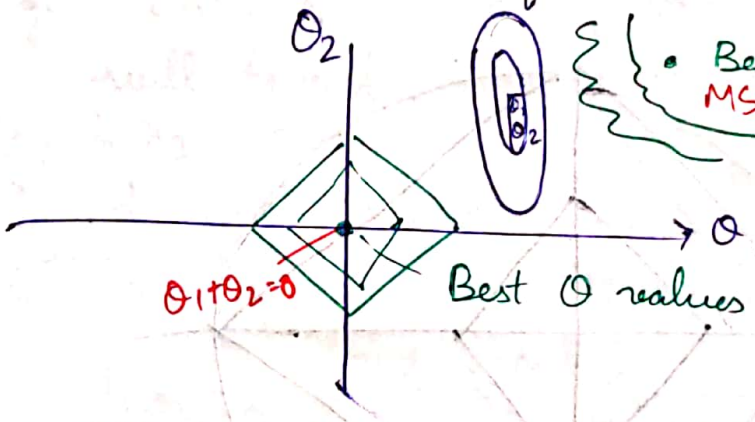
$\sqrt{|\theta_1|} + \sqrt{|\theta_2|}$ $10+0.3$ {It bothers least in reducing θ_1 & θ_2.



Non Regularized Cost Func.

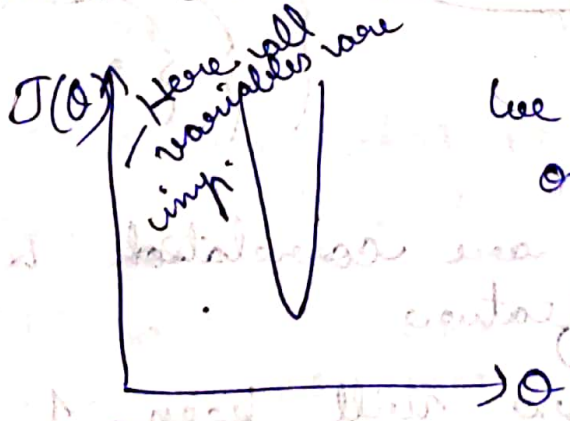
So, when MSE reduces, θ goes up
when MSE ↑, θ ↓

Best case scenario for theta & MSE

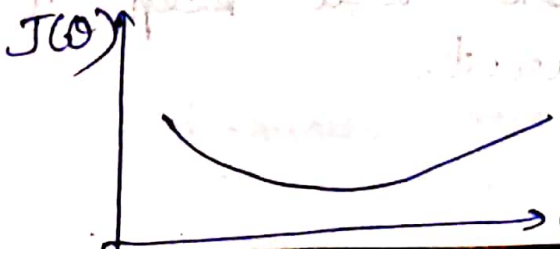


θ_2 = penalized more than θ_1, since prob. of θ_2 touching the end pt. is more

So MSE & θ cannot dec zero, so we can inc θ & MSE simultaneously.



We can do regularization here, since on red. θ, MSE shoot up very much so reg. cannot do much effect on such type of curves.



→ Here reg. will play more role θ↓, MSE is hardly changing

→ Scaling doesn't mean every curve will be same

→ λ is very tiny, contour won't expand, diamond will expand very much.

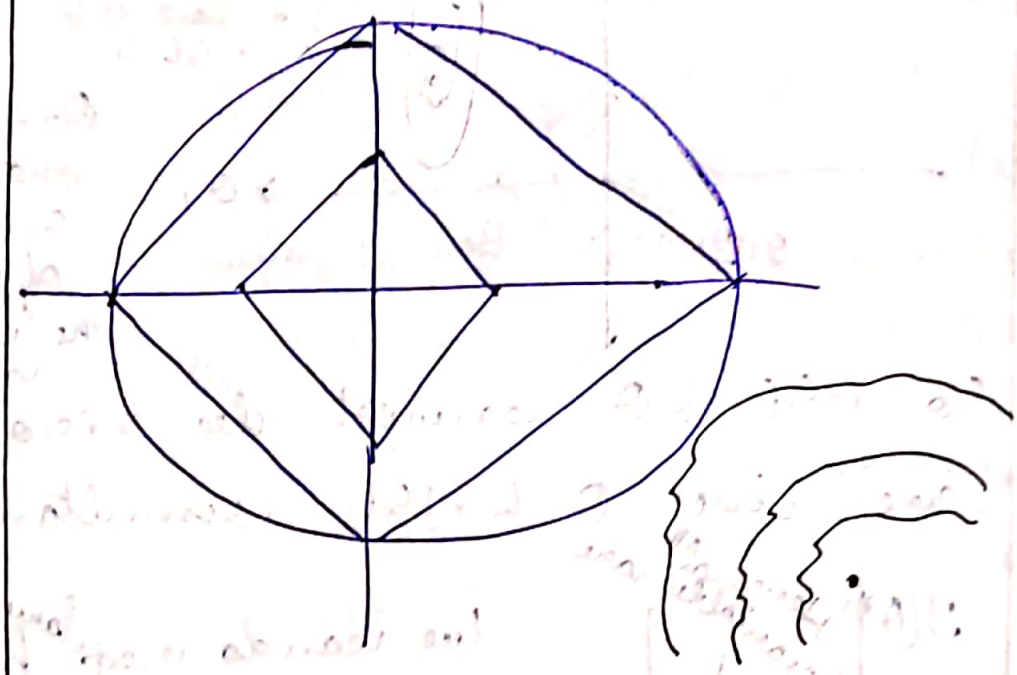
→ $\lambda \uparrow$ very large, it will say that MSE rod must come to diamond, it will touch diamond ine. Optimal pt. is (0,0)

ELASTIC NET

$$\frac{1}{2}\text{MSE} + \lambda_1 \sum |\theta| + \lambda_2 \sum \theta^2$$

$\lambda_1 \sum |\theta|$ → doesn't have impact large θ values

$\lambda_2 \sum \theta^2$ → Impact large θ values



→ When features are correlated & there are multiple features

- Lasso will keep 1, remove the rest.

Eg:- I have Room Size
and 5X RoomSize

Lets say I remove 5X RoomSize.

Carpet Area - Inside Room carpet area.
Built-up. = Takes area into the walls. with area
Super Build-up. = Net Area / Total Apartments.

→ Lasso will remove any 2 variables.

$$1000 \times CA + 50 \times \underbrace{SBU}_{\times 0.2 \times CA} + 100 \times \underbrace{BU}_{\times 0.2 \times CA}$$

→ The red. in MSE will be ↓, but Lasso will turn 2 of them to zero

→ Ridge will keep these variable

→ E.N will depend on λ_1 & λ_2. For val. of λ_1 it will remove all the (Lasso route) for val. of λ_2 it will keep the cluster. (Ridge route)

↓ Find cluster of var. which will preserve all cluster & keep all corr. cluster.

disadv. → Here, tuning λ_1 & λ_2 is difficult to tune.

AIC Akaike Info. Criteria } It is used to compare
BIC Bayesian " " } models.

BIC assume that one of them is a true model & try to rank in that.

AIC doesn't assume that anything is a true model { Not a single model are perfect }

" All models are wrong, that is each model will have some drawback"

BIC → pick one true model present in the cluster of models.

AIC → doesn't pick.

<Difference>

BIC	AIC
Stricter on nos. of parameter.	Not stricter. Rigid
Picks model that is simpler	→ Picks complex model.

→ Both checks R^2 & MSE.

→ AIC → $2K + \ln RSS$

BIC → $K \ln(n) + \ln RSS$

K = nos. of parameter

n = nos. of data pts.

If, $n > e^2$, then $k \ln(n) > 2K$

will be always greater, $e^2 = 9$, $e = (2.737)^2 \approx 9$

Soo, BIC is always practically greater than AIC.

has higher weightage on k, so it always choose simpler model

BIC → better in case of overfitting

AIC → " " " " underfitting

So, both are better but people tends more towards AIC since no model is good

---

If $n \downarrow$, then AIC ends up picking the better model.

When $n < k^2$, always go for AIC.

Text data, TF-IDF, features shoots up very much.

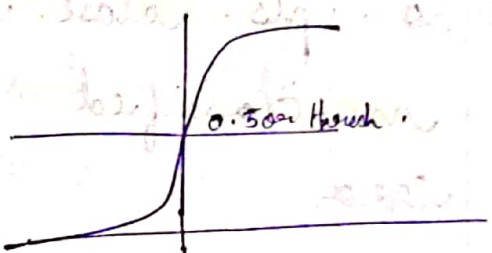
LOGISTIC REGRESSION

Sigmoid func:

$$\hat{y} = \frac{1}{1 + e^{-\theta^T x}}$$

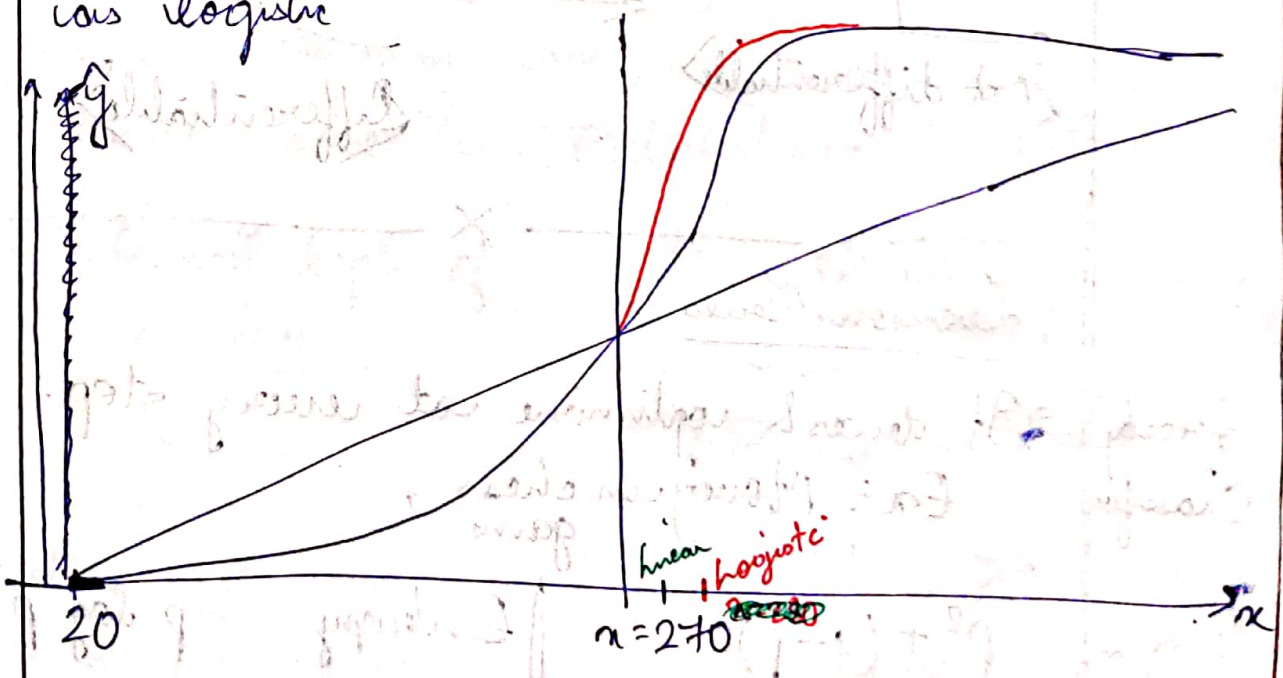
$\theta^T x \to \infty$, formula $\to 1$

$\theta^T x \to -\infty$, " $\to 0$



→ In log., even if a point is slightly mis-classified, it will penalize heavily.

→ But linear will not penalize as much as logistic



$\hat{y} = 0.002x - 0.04$ (linear Eqn)

when $x = 270$

$\hat{y} = 0.5$

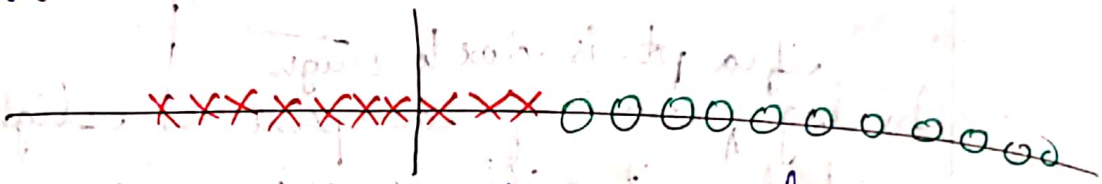
When $x = 280$

$\hat{y} = 0.52$ | $\hat{y} = 0.6$ (Logistic)

When $\hat{y} = 0.6$ | $\hat{y} = 0.85$

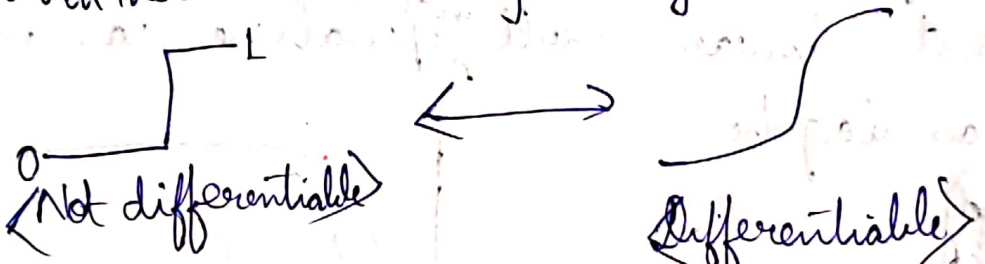
→ So it allows it to learn much faster. Since it induce much higher error in logistic than linear.

→ So, pts. close to origin & if slightly misclassified. Sigmoid induces much high error



(Check) Cont. function

When there is arbitrary change in slope.



Decision Trees

Greedy Classifier → It tries to optimize at every step.

Ex: Moving in chess game.

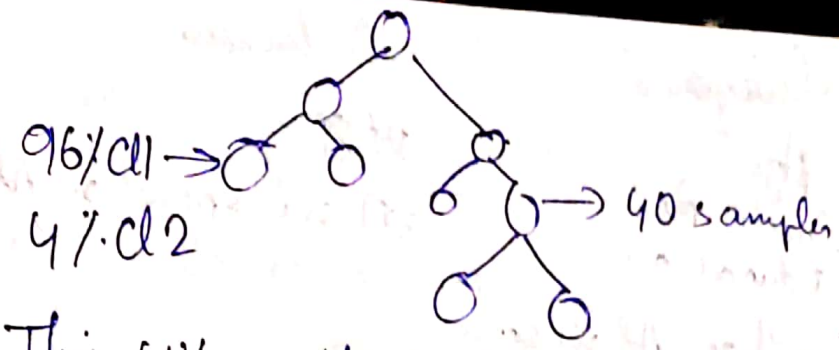
Gini Index $p^2 + (1-p)^2$

Entropy $-p \log p$

Info. gain

Chi Sq.

Variance Reduction



→ This 4% could be noise ; { Plz don't split when very few samples }
Becoz it try to reach homogenity.

→ 2 data pt. were identical then acc. is less than 100%.
Or else in every other case accuracy is 100%.

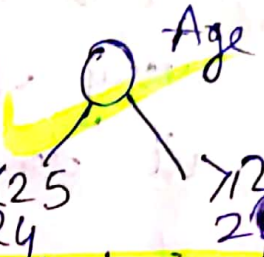
→ So, its imp. to put restrictions on tree to remove this overfitting.

→ some restrictions are;
① node size ? ② what is the I.G or Gini
③ Overall depth of the tree. If gain is minimal it wont split.

DT Vs ~~KR~~ Logistic Regression

① Can fit very complex func. Less likely to overfit.
② Robust to outliers less likely significantly change. We have to handle outliers
③ OK with missing values ③ More faster than D. Tree
Missing value will always go into the 'NO' part
④ Better with class skewed

Categorical	Continuous
1) Age	1) Age
2) State Resi.	3) Time spend on Ad.
3) Education	
4) Weather Ad is been clicked or not	



D. Tree → In cont. variable to put a boundary is a good idea, since, 24 years old person is similar in behaviour to 26 year old person.

Categorical Logistic Reg

M = 0
Fem = 1

Either 0.6×1 or 0.6×0 → THETA VALUE

Doesn't Make Sense

(State code) 20 - UP, 1 - Andhra

0.5×20 (No Sense) ← (0.5×1)

Cont.

$0.37 \times$ Age

$0.5 \times$ Age

Age	Age
12	13

Similar

Age 28 → or 39×0.5, 41×0.5, 59×0.5

Make Sense: 19.5, 20.5, 29.5

Similar

→ logistic suitable to cont. variable.

Log R would work with Cat. but it would be difficult for it to optimize the cost func; whereas in case of DT it is very easy for model to optimize those variables.

D.T can build any type of non-linear boundary very very quickly.

Naive Bayes is very very very very - - - less likely to overfit.

Better with Imbalanced Dataset

→ Class imbalance is handle at lower levels

Suppose, Class 1 - 90% Class 0 : 10%.

for above case, first Class 0 will split at one, & then Class 1 will split

So, Depth (↑) - we need to use the depth starting

→ Robust to outliers, outliers → usually goes down at last splitted

Depth (↓)

at end so we need to reduce the depth.

D.T	Logis R
Better with Categorical var.	Better with Conti. Variables

After Break

When output variable is continuous

Gini Index will not work if

y = weight

x is Height, Gender or any categorical var

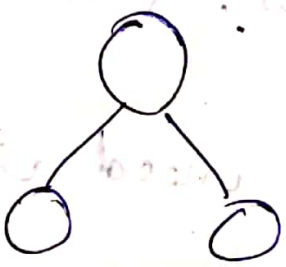
→ Gini Index will not work

Reduction in variance: If I see weather one feature is reducing the variance or other feature is reducing the variance or not

If checks if variance is reduced by large amount

Ultimate aim: There should be zero variance

Classification (y = cate.)



→ Prob. Statement: how to decide from which values to split

Where to create boundary when we want to analyse the time spend on social n/w.

values go from 0 ---------- 10^6

??

Gini Vs ChiSq. Vs Entropy

Faster.

log is slower, compu expensive computation

Result for all three will mostly be same.

Ensembling

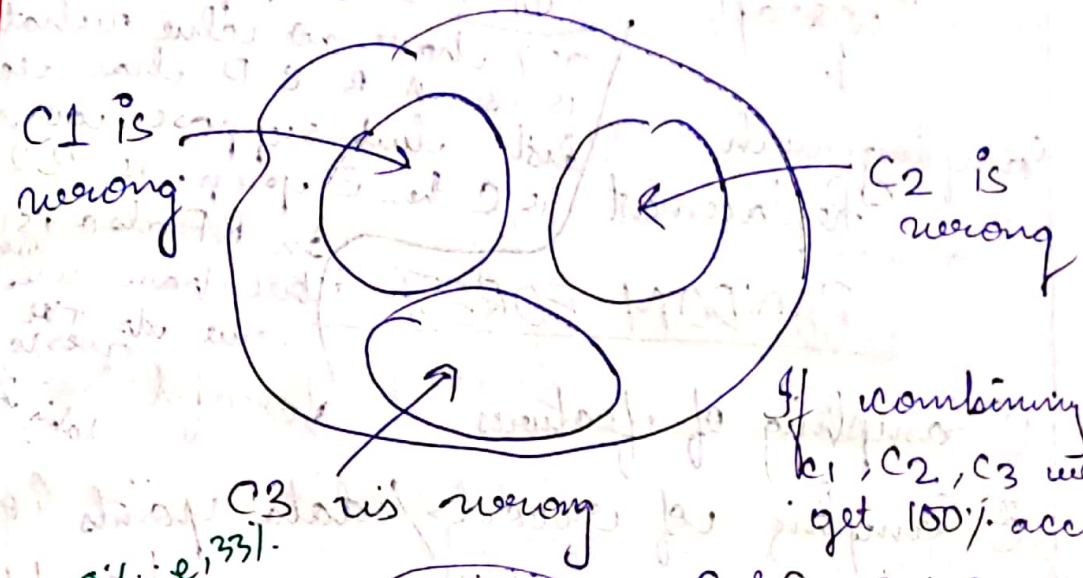
Accuracy

weak classifiers

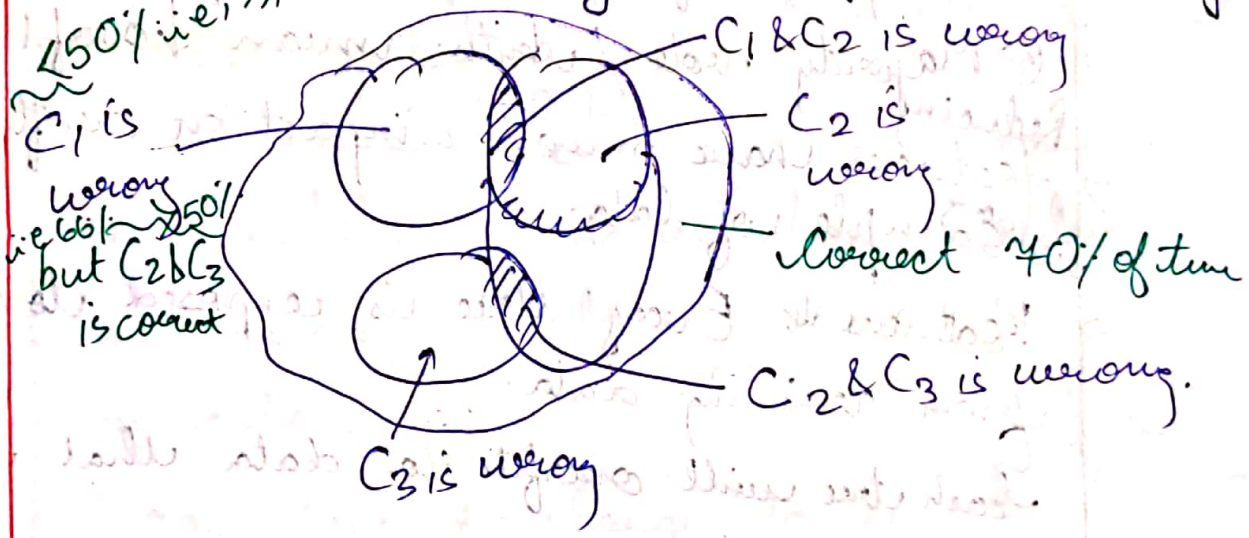
0 — 0.5 — 1

Weak classifier : better than random guess

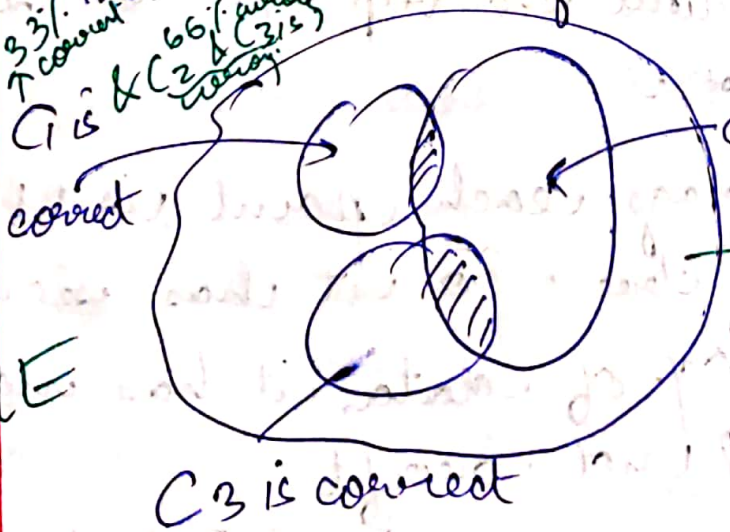
~~Ex: If tossing a coin 5 time then $p > 0.5$~~



If combining all h_1, C_2, C_3 we will get 100% accuracy.



When intersection takes 50% of area, then the classifier is not good.



At least 2 needs to be correct which is shaded area for the ensemble classifier to be correct.

Imp, Imp ENSEMBLE COND

Cond: At least each classifier should be correct above 50%.

BAGGING

Bootstrap → Sampling with Replacement

Aggregation

RANDOM FOREST

(1) Sampling of features ↓ overfit

(2) Sampling of records / data points ↑ overfit

(3) Majority vote / avg. mean ↓ overfit

Reducing

→ features have much impact on overfitting than reducing sample of data.

→ Features ✱ Every tree is exposed to something from training data.

Each tree will overfit to data that it has seen.

→ If 20% of data has been used by 1 D-Tree 1 data pt. will end seen in the 20% of the data. ✓

On an average each point is exposed to 20% of tree. So it has seen 20% of data & 80% of data it has not seen so it will not overfit.

→ If we are exposing 80% of data then we are also reducing the effect of 2nd point that is we are

(#3) Explanation

In KBC, suppose audience poll 10% of pop. knows answer 90% have no clue what the ans is so A B C D has each 22.5% dist. but suppose correct ans is C then C prop. is 32.5% rest 22.5

In RF also, suppose 50 trees have rule, 25% have clue its tree, so it goes to tree taking less

So, in this case also it well manage overfitting.

Bagging loose out on speed, interpretability, reduces overfit

	Rand. Forest	vs	Decision Trees
①	Less likely to overfit		Interpretable
②	Handles Noisy labels		Speed (Especially prediction speed)
③	Tells Feature Importance (It check at what thigh feature is present in each D.Tree)		Less likely to underfit

Noisy labeles: Fraud detection problems have noisy labels.

Transaction Fraud – Small lag of time (Anomaly Detec. or Sup. learning)

Loan Fraud - Large lag of time

Lets say we took last 6 months of data (May – Oct) we give to model. Here we mark which are fraud or not. On 25 oct. fraud happens, but in data - lease 25 dec. has not fraud., so it will show not fraud on 25 oct in model

→ None Deep learning do this. Auto Encoders we dont use labeles.

→ Becoz of bagging 20% data we are exposing, so it is handles noisy labels.

→ RF is worst since we are reducing features
RF will underfit worst.

→ Ex, 10000 data pts & 3 features.
RF will take 2 features & build D.T
n nos. of times but none of the times
D.T is able to predict the labeles.

WHEREAS, D.T will take all 3 features
10,000 data pts & will be able to
predict the labels.

BOOSTING

Suppose we have bunch of three classifiers $h_1, h_2, h_3, \ldots$

$h_1, h_2, h_3 \ldots \rightarrow +1$
$\rightarrow -1$

$h(x) = \text{sign}(h_1(x) + h_2(x) + h_3(x) + \ldots)$

What if I boost all misclassified datapoints in our next classifier.

$\sum w_i = 1$, lets scale the weights so that its better to calculate.

$\epsilon = \sum_{\text{i E Errors}} w_i$

Now boost stronger classifiers

$h(x) = \text{sign}(\alpha_1 h_1(x) + \alpha_2 h_2(x) + \alpha_3 h_3(x) + \ldots)$

Steps

1. Starts with
$w_i = \frac{1}{N}$ → Pick h_t that minimizes ϵ_t

$w_i = \frac{1}{N}$ → Pick h_t that minimizes the ϵ_t

Ada Boost

Calculate w_{it} ← Pick α_t

$$w_i^{t+1} = \frac{w_i^t}{Z} e^{-\alpha_t h^t(x_i) y_i}$$

$$\alpha^t = \frac{1}{2} \ln \frac{1-\epsilon_t}{\epsilon_t}$$

$e^{-\alpha_t(+1)}$ when pred. is same as model

when $h^t(x_i) = y_i$

when $h_t(x) = 1, y_i = 1$ or $-1, y_i = -1$ ↑ if correct

$$w_i^{t+1} = \frac{w_i^t}{Z} e^{-\ln\sqrt{\frac{\epsilon_t}{1-\epsilon_t}}}$$

$$= \frac{w_i^t}{Z} e^{\ln\sqrt{\frac{1-\epsilon_t}{\epsilon_t}}}$$

when opp. pred. then $e^{-\alpha_t(-1)}$ ↑ if wrong

$$\Rightarrow \frac{w_i^t}{Z}\left(\sqrt{\frac{\epsilon_t}{1-\epsilon_t}} + \sqrt{\frac{1-\epsilon}{\epsilon_t}}\right) = 1$$

$$\Rightarrow \frac{1}{Z}\left(\sqrt{\frac{\epsilon_t}{1-\epsilon_t}} \sum_{\text{correct}} w_i^+ + \sqrt{\frac{1-\epsilon_t}{\epsilon_t}} \sum_{\text{wrong}} w_i^-\right) = 1$$

$$\Rightarrow \boxed{Z = 2\sqrt{\epsilon_t(1-\epsilon_t)}}$$

$$\sum_{\text{correct}} w_i^+ = \frac{1}{Z}\sqrt{\frac{\epsilon_t}{1-\epsilon_t}}$$

$\sum w_i^+$ correct $= (1-\varepsilon)$ $\quad$ $\sum w_i^-$ wrong $= \varepsilon$

So, $z = \frac{\sqrt{\varepsilon_t}}{\sqrt{1-\varepsilon_t}} \times (1-\varepsilon_t) + \frac{\sqrt{1-\varepsilon_t}}{\sqrt{\varepsilon_t}} (\varepsilon_t)$

So we get $z = 2\sqrt{\varepsilon_t(1-\varepsilon_t)}$

$w_i^{t+1} = \frac{w_i^t}{z} \sqrt{\frac{\varepsilon_t}{1-\varepsilon_t}}$ if correct

$= \frac{w_i^t}{2\sqrt{\varepsilon_t(1-\varepsilon_t)}} \sqrt{\frac{\varepsilon_t}{(1-\varepsilon_t)}}$

$= \frac{w_i}{2(1-\varepsilon_t)}$

if wrong

$\frac{w_i^t}{2\sqrt{\varepsilon_t(1-\varepsilon_t)}} \sqrt{\frac{1-\varepsilon_t}{\varepsilon_t}}$

$\frac{w_i^t}{2\varepsilon_t}$

$\sum_{\text{correct}} w_i^{t+1} = \frac{1}{2}$, w_i when correct is $1-\varepsilon_i$

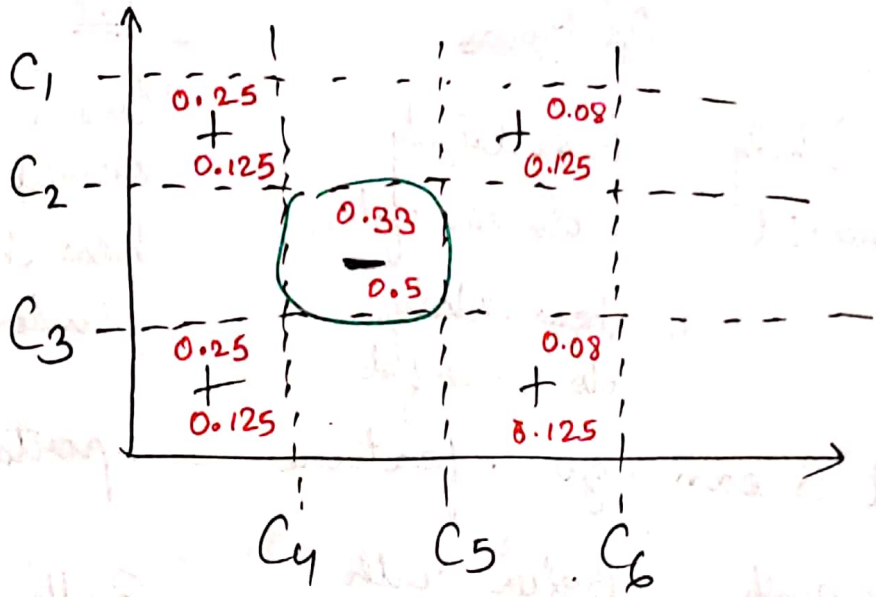
So $\frac{1-\varepsilon_i}{2(1-\varepsilon_t)} = \frac{1}{2}$

& $\sum_{\text{incorrect}} w_i^{t+1} = \frac{1}{2}$, w_i when incorrect is ε_i

So $\frac{\varepsilon_i}{2\varepsilon_t} = \frac{1}{2}$

This works if if we have more 50% correct data so that it weights correct as $\frac{1}{2}$ & incorrect as $\frac{1}{2}$

→ Suppose 80% of correct & 20% of incorrect data is there, so it gives 80% of desired & rest 20% ,, ,, ,, ,, $\frac{1}{2}$



Lets say, I have Some classifier $h(x_i)$ GBM's

$$y_i^t = h^t(x_i) + err^t$$

$$err^t = h^{t+1}(x_i) + err^{t+1}$$

$$err^{t+1} = h^{t+2}(x_i) + err^{t+2}$$

⋮

At some pt., error becomes very small.

$$y_i = h^t(x_i) + h^{t+1}(x_i) + h^{t+2}(x_i) + \cdots$$

→ So, in GBM's it tries to correct the error from the previous one.

RF	AdaBoost	GBM
less likely to overfit	less likely to overfit less likely to underfit	More likely to overfit less likely to underfd

All 3 can give feature importance

works with noisy labels Since we use D.Tree, so it can handle Missing Value, Outliers	Better with class skew	Better with class skew

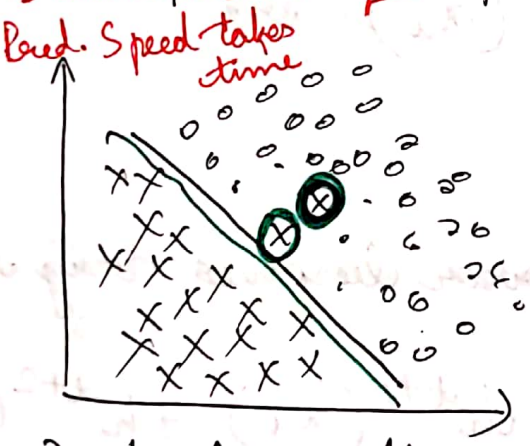
Depends on what type of underlying classifier we are using.

Training Speed when parallel

Prediction Speed

Prediction Speed.

< logistic is faster than all >



● → Ada boost

Overfitting is when it doesn't perform well on unseen data. But since we do seperate circle, Ada Boost does well on Test or unseen data.

XGBoost.

GBM + Regularization + Speed.

↓

Reduces overfitting & its fast.

Problem with boosting:

We need lot of hyperparameter tuning.

Day 3: 21-Dec-19

Till now we have studied ~~Generative~~ Discriminative model, today we will learn about Generative model.

⇒ In Generative Model we will learn Joint probability $P(c,d)$

⇒ Joint prob of c and d is

$$P(c,d) = P(d/c) \cdot P(c) = P(c/d) \cdot P(d)$$

$$P(c/d) = \frac{P(d/c) \cdot P(c)}{P(d)}$$

$$\text{Max}(P(c/d)) = \text{Max}(P(d/c) \cdot P(c))$$

$P(d/c)$ = Lets say I have 5 features

$$P(x_1, \ldots, x_5 / c)$$

Joint prob of all the features given the class

Now expanding this thing: $P(x_5/c)$

$$= P(x_1, x_2, x_3, x_4 / x_5, c) \wedge P(x_1, x_2, x_3 / x_4, x_5, c)$$

$$P(x_4/x_5,c) P(x_1, x_2 / x_3, x_4, x_5, c) \vee P(x_1 / x_2, x_3, x_4, x_5, c)$$

$$P(x_3/x_4,x_5,c)$$

This is very difficult to compute since the probability gets smaller and smaller.

$P(x_2/x_3, x_4, x_5, c)$

Assumption ⇒ So, we assume that all the features are independent of each other given the class c.

Complexity of Naive Bayes if there's no assumption

$O(|X|^n \cdot |C|)$ parameters.

So, after assumption we can write as, $P(x_n/c)$

$$P(x_1, \ldots, x_n | c) = P(x_1|c) \cdot P(x_2|c) \cdots P(c)$$

$$c_{map} = \underset{c \in C}{\operatorname{argmax}} \; P(x_1, x_2, \ldots x_n | c) P(c)$$

$$\underset{c \in C}{\operatorname{argmax}} \; P(c_j) \prod_{x_i \in X} P(x|c)$$

word positions in that docs.

$$c_{NB} = \underset{c_j \in C}{\operatorname{argmax}} \; P(c_j) \prod_{i \in \text{positions}} P(x_i | c_j)$$

1st Attempt: Maximum likelihood estimates

Simply use the frequencies in the data.

$$\hat{P}(c_j) = \frac{\text{doc\_count}(C = c_j)}{N_{doc}}$$

$$\hat{P}(w_i | c_j) = \frac{\text{count}(w_i, c_j)}{\sum_{w \in V} \text{count}(w, c_j)}$$

→ Zero prob. cannot go away, its a big challenge, so people use smoothing

What if we have seen no training doc. with the word fantastic & classified in the topic +ve.

$$\hat{P}(\text{"fantastic"} / \text{positive}) = \frac{\text{count}(\text{"fantastic"}, \text{positive})}{\sum_{w \in V} \text{count}(w, \text{positive})}$$

(1) laplace smoothing: Just adding one in the numerator & the denominator.

vantage
1. Very Very Fast.
2. Low Storage Requirements.
3. Extremely robust to irrelevant features.
(least likely to overfit than neg., DT, Boosting, less than Random Forest, its a benchmark)

Eg: Weather somebody going to cancer or not
" it will or not. Yes on that particular day

Sol: Both cond? has no relation

So, $P(R=Y/C=\text{Cancer}) = 0.3$ ~~$P(R=N$~~

$P(C=Y/C=\text{Not Cancer}) = 0.3$

So, weightage for both the class will be equal, so no overfitting.

4. Both of the features are given equal importance but DT suffers from fragmentation in such cases - especially if little data.

Eg: Weather an email is spam or not-spam. They will just repeat the subject. So, if a particular feature is imp. just duplicate it & give 6 features if 5 features is there - Weather or not there is 100% correlation, it will work

best by giving extra importance to that particular features whereas all other ML model will struggle like correctly

→ It works really well in very high dimensional datasets, so high dimen. datasets like text like In millions docs, if I take 10K for bag of words & do bigrams & unigrams. So that I have 1 lakh features. Its very difficult to figure out right params when there are lot of features.

It learn lot better & lot faster, less likely to underfit (Discriminative)

It generalize very very well that is it will never overfit

→ Predicting stock price - Difficult prob. (Use discrim classifier)

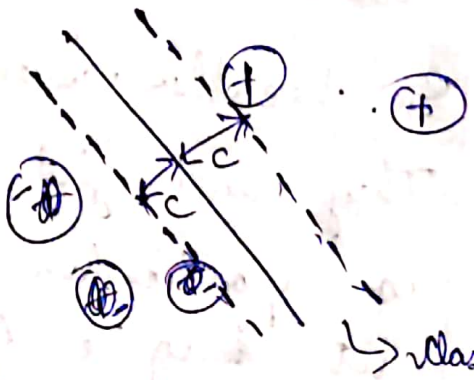
→ When to classify dog - Simpler prob. (Use generative classifier)

SVM

① The distance between points & boundary is as high as possible as in logistic there is no such rule.

$\vec{x}$ = data pt.

$\vec{w}$ = ~~the vector that width on the~~ vector that allows me to determine the dis. b/w boundary vector $\vec{x}$ & decision boundary.



$\vec{w} \cdot \vec{u} - c \geq 0$

$\vec{w} \cdot \vec{u} + b \geq 0$, here b is $-c$ (Constant)

→ For logistic the objective is to just create a boundary or just seperate the data.

Now seperating +ve and -ve data points

$(\vec{w} \cdot \vec{u} + b) \geq 1$ → For +ve

$(\vec{w} \cdot \vec{u} + b) \leq -1$ → For -ve

Combining both the eq^n :

$y_i(\vec{w} \cdot \vec{u} + b) \geq 1$

↑ Outcome either +1 or -1

$\vec{x_+} = (1-b)\frac{\vec{w}}{\|w\|}$ $\qquad$ $\vec{x_+} - \vec{x_-} = \frac{2\vec{w}}{\|w\|}$

$\vec{x_-} = -(1+b)\frac{\vec{w}}{\|w\|}$

So, we want to Maximize the Margin,

by Maximizing $= \frac{1}{2}\|w\|^2$ with

$y_i(\vec{w} \cdot \vec{u} + b) - 1 \geq 0$ as my constraint

→ Lagranges say I can combine these 2 eq^n & minimize this

$\frac{1}{2}\|w\|^2 + \alpha\left(\sum y(wu + b) - 1\right)$ If not violated

Margin $\qquad$ Error part

$\frac{1}{2}\|w\|^2 - \alpha($ " $)$ If violated

I want to min $\frac{1}{2}\|w\|^2$

I say that the error is not violated. What is if its not violated then it will always be zero.

If error is -ve then ans is -ve.

α is a weightage which give weightage to each point.

If it is violated then we will multiplied by a weightage

If α is very very large, it will never get violated

$$\frac{dL}{\partial \vec{w}} = 0$$

$$\vec{w} - \Sigma \alpha y \vec{x} = 0$$

$$\vec{w} = \Sigma \alpha y \vec{x}$$

$$\frac{dL}{\partial b} = \Sigma \alpha y = 0$$

$$L = \frac{1}{2} \sum \alpha_i y_i \vec{x}_i \sum \alpha_j y_j \vec{x}_j$$

$$- \sum \alpha_i y_i \vec{x}_i \sum \alpha_j y_i \vec{x}_j$$

$\to 0$

$$- b \sum \alpha_i y_i + \sum \alpha_i$$

till now I did for linear separable or linearly class.

$$\leftarrow L = \sum \alpha - \frac{1}{2} \sum \sum \alpha_i \alpha_j y_i y_j \boxed{\vec{x}_i \cdot \vec{x}_j}$$

Now, I can build a kernel func?, so that I can transform it to new dim, so I need to transform the dot products to the transform space.

→ I new dot prod. of every comb. of our data

$$K \Rightarrow (\vec{u} \cdot \vec{v} + 1)$$

→ It is better in high dim. data (like text data)

So, kernel func. does this. On inc. dimen. it will not overfit since its method is to maximize the margin

---

→ PCA always do linear comb. of features that why we use kernels in PCA

Two dim. proj.

Only PCA Kernel PCA

Disadv -

① Slow, need to figure out right kernel, need to do tuning.

Neighbourhood based Prediction (KNN)

1) It is computationally expensive when 1 Million data pts is there.

2) So, there are bunch of tree steps. KDTree, Vantage Point VP Tree. These tree allows to cal. yrt. ~~soon~~ quickly

3) So for mapping faces to see weather a face is matched or not.

Eg: In Shadi.com, criminal faces not allowed, victor is not allowed. If face matches to one if they say give id proof) it goes to CRO

4) It is still slower but with this VP trees & all we can make it faster than RF & SVM.

5) It is very very quick to underfit. Depends on K. IMP.

If to check the temp. tommorow in Mumbai. If I keep K = 1 Million, then it practically gives the avg. of the entire dataset (Underfit)

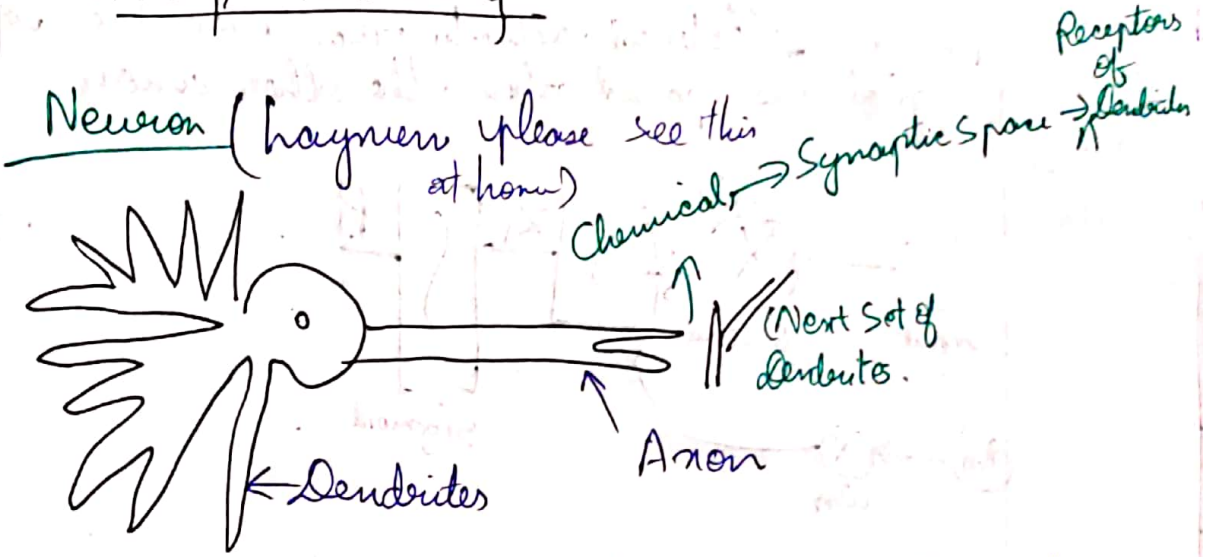
If K=1, it will take immediate neighbourhood point. (Overfit)

K = 3, 4, 5 are good beyond that it will underfit

KNN ~~voronoi~~ Voronoi diagram ??

Deep learning

Neuron (Learners please see this at home)

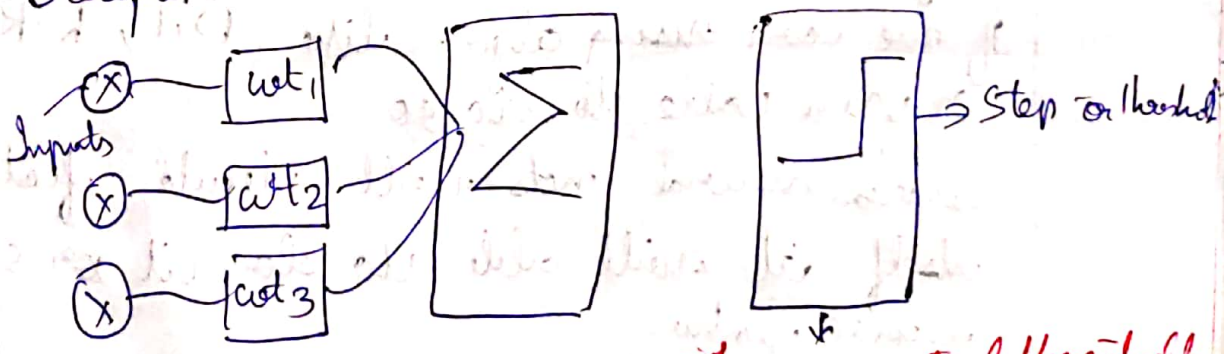


2) If a threshold is not cross no signal will pass
"All or None

3) So, there are dendrites which are closer to a particular Axon.

Cumulative Influence: Multiple neuron triggering
So multiple neurons cumulatively influence a reaction.

4) Synaptic Weight: A particular dendrites will have certain amt. of wt. for a particular receptor.



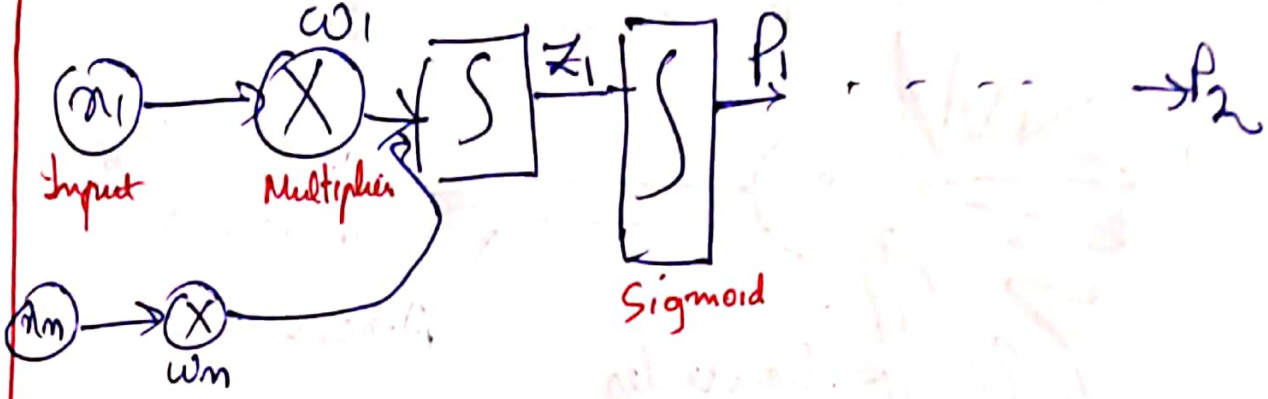
This was not differentiable

So they replaced with sigmoid.

Refractory period: After a signal is passed it need some time to for other signal to pass

In neural nets no such concept is there

Axonal Bifurcation: That is if one axon more electrical activity than in one via axon signal is passed than the other axon.



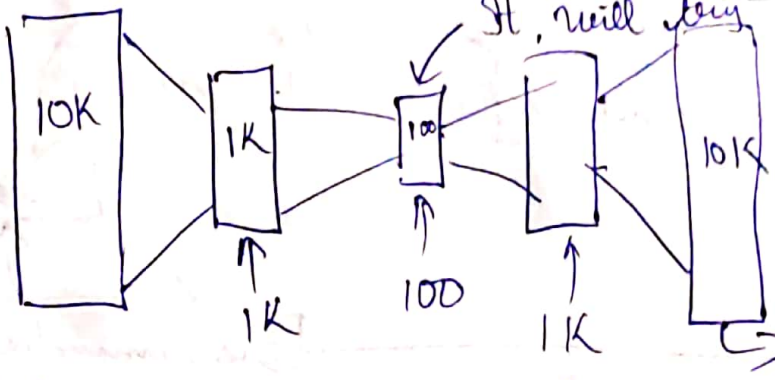
$$\frac{\partial P_2}{\partial w_2} = \frac{\partial P_2}{\partial z_2} \cdot \frac{\partial z_2}{\partial w_2}$$

$$\frac{\partial P_2}{\partial w_1} = \frac{\partial P_2}{\partial z_2} \cdot \frac{\partial z_2}{\partial P_1} \cdot \frac{\partial P_1}{\partial z_1} \cdot \frac{\partial z_1}{\partial w_1}$$

Linear in ~~Width~~ Depth
Quadratic in ~~Depth~~ Width

Ex. 1) In Shaadi.com suppose a person is from punjab & looks wise he will have a tough time if he leaves in Chennai.
If we were using algo. like D.T, L.R, SVM. It can calle to do so.
Whereas neural nets will build features by itself it will able to do it or solve the above feature.

Auto Encoders



It will try to figure out what the pixels is & it tries to learn it

It will try to draw it or pred. the picture

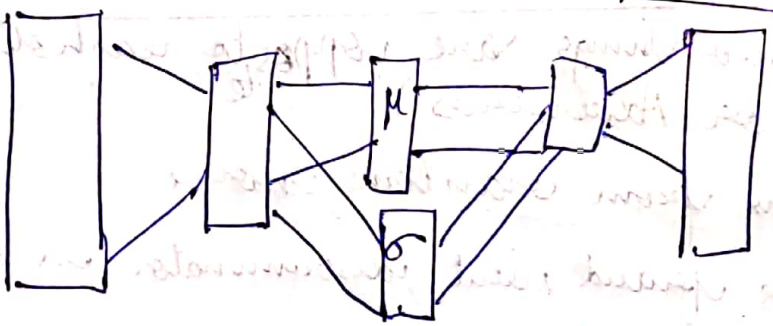
If I give 10K values, then give me 10K value at output

The problem here is it cannot overlearn all the values.

Denoising Autoencoder (DAE)

1) It will reconstruct from partial data.
2) Make model robust to words noisy data

VAE (Variational Autoencoder)

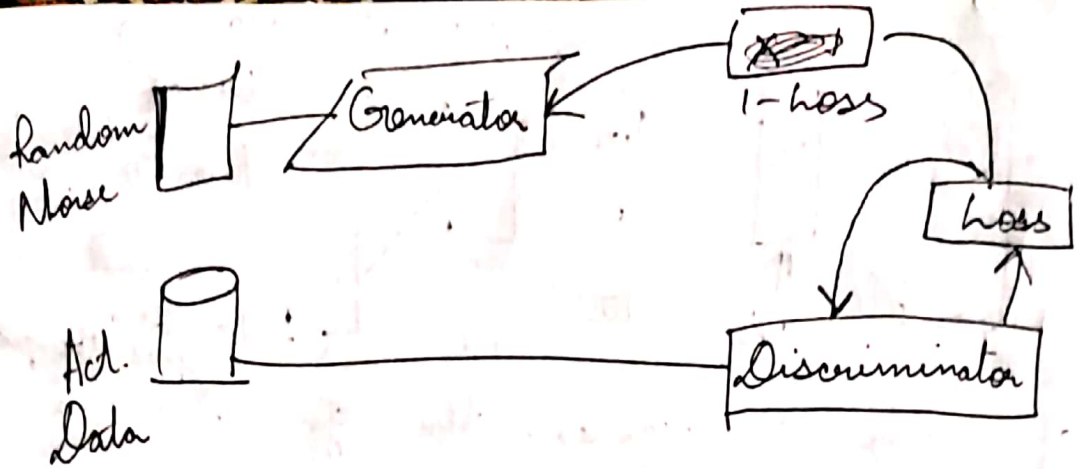


Used in Anamoly Detection :

How? If S.D falls beyond a particular S.D like 3.. 5., we will reject it.

⇒ It check mean & SD & then learns from the mean & SD generated.

GANS (Generated Adversarial N/w):



Lets say I am generating faces (100x100) → give this to discriminator, & I tell to him that if you predict that its a actual then its a wrong.

Imp

Discriminator will be better in segregating which is real & non-real.

Generater will be better if it produce data which is very very close to real.

Adversarial: Two things are opposite to each other (two competitions or Adiversaries)

Generator learns from random noise

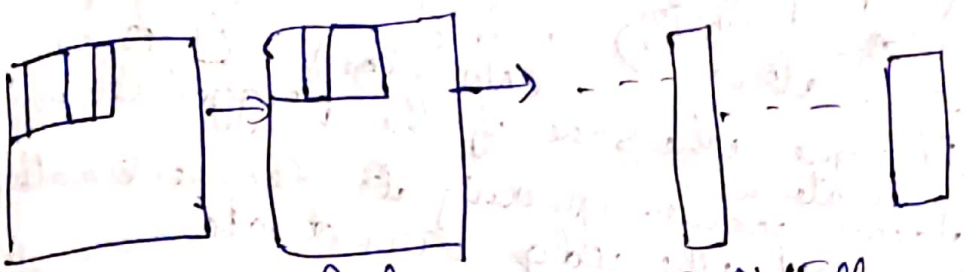
Generator make fraud, but discriminator is able to discriminate b/w fake & real.

Challenge: If one model is better than other (like Dis. is very very powerful than Generator) then the system stops learning.

Powerful Discri & able to
Generator will not learn since everytime loss will be large

Powerful Generator
Discriminator could be able to distinguish b/w fake & real & moreover loss is same everytime.

CNN



Conv. Pooling FCN (Fully connected n/w)

Here there is func. that is applied to every matrix of our data.

Like detection edges, we get certain output

Then we apply pooling (typically max pooling)

→ What will happen if pooling layer is removed.

It keep what's imp.

Prevent Overfit.,

There is significant impact on computational time.

→ Conv. will detect edges.

→ Why Max pooling is used?

Pooling by itself is useless.

Justify what's happening in Conv., Each conv. is detecting the prominent parts of image.

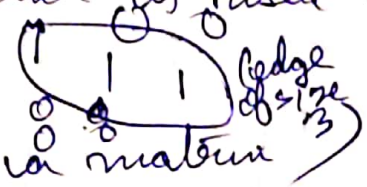
8 2 → 8 prominent features is detected

3 4

By itself its useless, so its not the 1st layer everytime.

→ So pooling pick most prominent pieces from the conv. layer.

→ Different kernel is used by the model to detect the edge.



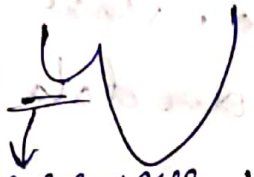
→ Kernel is in matrix

→ It will give the size of the biggest edge given any matrix

→ Once it is done some pooling is done decides whether to convey forward the edge or not etc.

Dropout layers: (to reduce overfit)

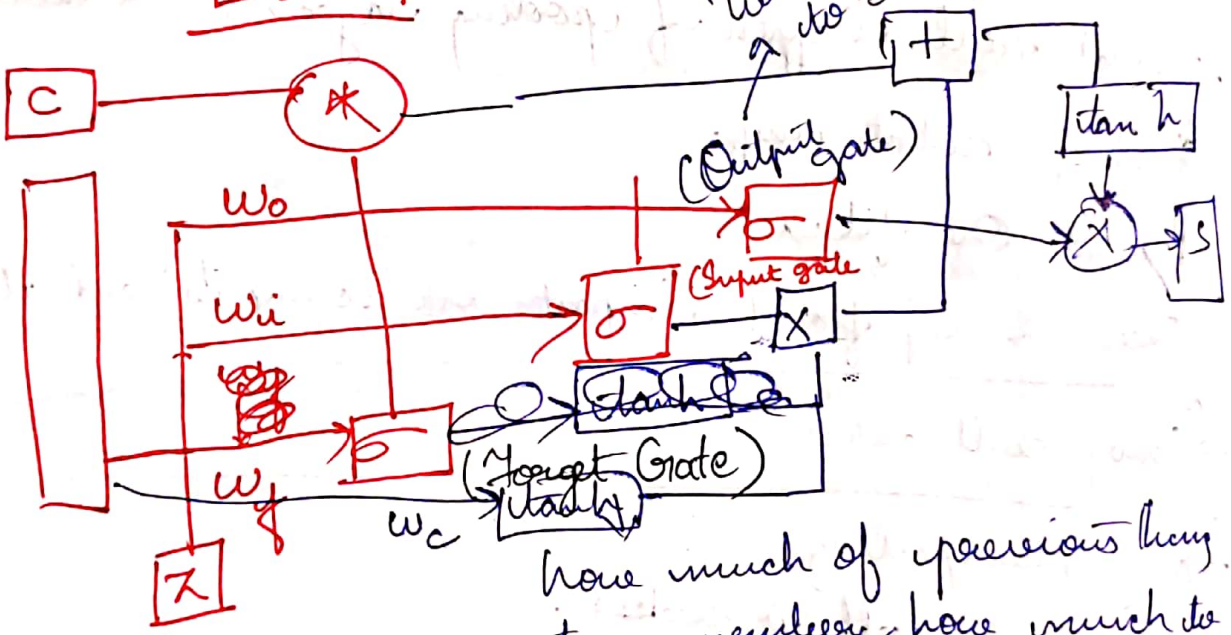
Neural Network



→ Dropout layers try to remove local optimas.

→ Enable to achieve global optima.

LSTM



how much of previous thing to remember, how much to forget.

The food is good but ... (forget things)

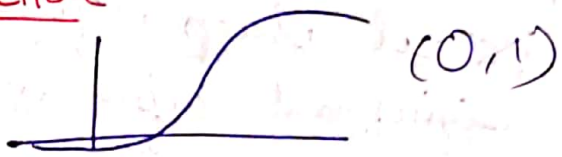
; -- (semi colon forgot)

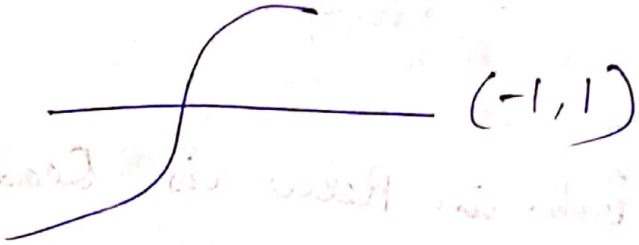
I/P gate: What from current world to be remembered.

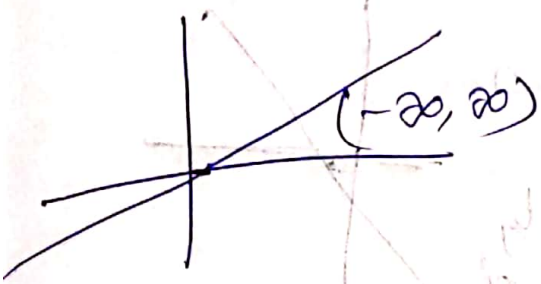
Q. When forget Gate is removed?
It provided the ability but this doesn't mean that it will forget. Its just a architecture. Wheather a model uses all of that depends on the type of training Data.

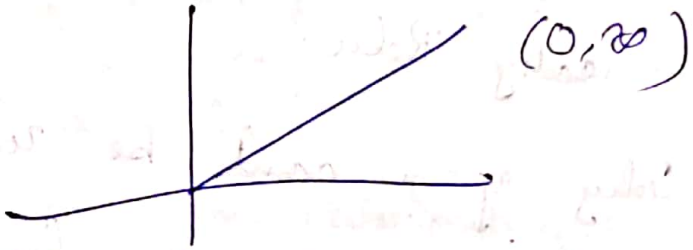
Activation Function

Sigmoid
Tanh
Relu
LRelu
PRelu
Selu



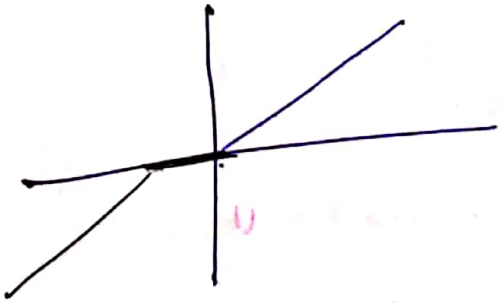




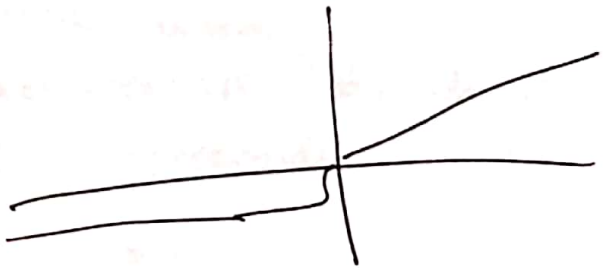


LRelu if +ve then its x
-ve then its 0.1x

PRelu (Probalistic Relu)



Selu



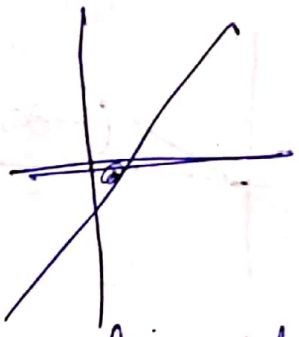
Vanishing Gradient

Sigmoid & Tanh } Squashing function } When squashing happens when we do back propagation for values 0.5 < 1.

Because of it there is no significance difference because it keep squashing, so there is no significant difference when it kept squashing.

Prob. in Relu is Dead Neuron, so we use Leaky Relu.

(1) Why $y = x$ can't be used?



There is no non-linearity in the data. The purpose of deep learning is to include non linearity in the data.

28-Dec-19 (Recommender Systems)

1. They are need to be personalised.
2. It needs to provide choice to the users.
3. They are inherently ranking system, it is important. Eg: It will present result in a ranking fashion (e.g., Google, Youtube, Netflix) [Top to bottom OR left to right]

How sorting is done

1) Sorting Phase: Sorting is done based on the ancestral post in Facebook. In Facebook.
2) Formula Phase: It gives weightage to a topic, friendship. It then sorts by this weightage.
3) Model Phase: Developing formula for every person is difficult. There comes a D. Scientist to build a Model.

The model will predict how likely a people will comment or like the post.

→ There was a problem in the model like birthday, anniversary but it does not mean that we want to see that.

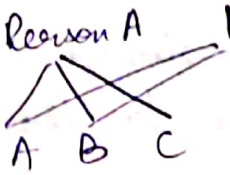
3) Creating Multiple Model & combining the models.
Google Ad: One model for how likely a user will click the ad.
Other model for a user to stay at least for 60 seconds.

4) Micro-personalisation Phase

Types of Recommender system

1) Popularity based
2) Collaborative
3) Content
4) Hybrid

Collaborative



(I it will recommend item 'C' since behaviour wit is similar for A & B

Content based

- It will recommend item similar to the item user likes
→ Ex: If one likes star trek movies, I will recommend star trek movies.

→ A/B testing is used to check current Rec. system with the new R.S, where current is exposed to 50% & rest new is exposed to 50% where we will see where we are getting more revenue

Types of Data to be used:

1. Explicit Data :- (User has given directly)
   - Customer Rating
   - Feedback/reviews
   - Demographics
   - Ephemeral Needs (I watched chinese movies)
2. Implicit Data :- (Infer something based on the customer activity)
   - → Purchase history
   - → Click or Browse history

Collaborative

Condition: 1. Many people need to participate.
2. Must have efficient algo.
3. There must be a simple way to express

Implicit Feedback

- Time spend viewing the product
- Nos. of times the product is received
- Products in wish-list
- Products brought

Explicit Feedback

- Rating given to a product
- Comments left for a product {This is better}

Explicit data is very less. Since people don't give rating everytime. It is inaccurate.

Mostly Implicit data is used, it depends on company based on the amt. of implicit or explicit data we have. We will give weightage to implicit as well as explicit.

Strong Generalization vs Weak Generalization

→ If all test set is present in train set, it is weak generalization.

Here uniqueness is user-item pair.

→ But if none of the users are present in the train set then its a strong generalization

---

→ Its important to see your results on both kind of test sets.

-ve Sampling

1) When we assume negatives we have to use this
Eg: If I show you 10 item from my system, I clicked on item 3 & item 5. I will say there is true for 3 & 5 & I assume that there is -ve for 1, 2, 4, 6, 7, 8, 9, 10. So here I will sample few these nos.; since I don't know weather it is actually -ve or not
2) So all +ves taken & sample of -ves are taken

Techniques in use: Similarity Index

Approach:

1) Represent a customer as an N-dimensional vector of items.
2) Vector is +ve for purchased or positively rated items and negative for negatively rated items

3. Based on cosine similarity : find similar customers/users, bet. A, B or A, C or, A, — or

$$\text{similarity}(\bar{A}, \bar{B}) = \cos(\bar{A}, \bar{B}) = \frac{\bar{A} \cdot \bar{B}}{|\bar{A}| \cdot |\bar{B}|}$$

4. Generates recommendations based on a few customers who are most similar to the user.

5. Rank each item according to how many similar customers purchased it. (Taking top 10 or 100 customers)

Problems:

1) Computationally expensive: $O(M \times N)$ is the worst case
   M = nos. of customers — if M is in millions
   N = nos. of items — N is in billions
   Not feasable

2) Dimensionality reduction can increase the performance. But, also reduce the quality of the recommendation.

3) For very large data sets, such as 10 million customers and 1 million items, the algorithm encounters severe performance and scaling issues.

So people started using Clustering Methods.

Approach

1) Divide the customers into segments & treat as classification problem
2) Assign user to the segment containing the most similar customers.
3) Users the purchases and ratings of the customers in the segment to generate recommendations
4) This method have better online scalability and performance. Implicitely clustering happened based on ratings

Problems:

1) Quality of recommendation is low.
2) To improve quality, it needs online segmentation, which is almost as expensive as finding similar customers using collaborative filtering.

DBSCAN – fast (anomaly detection) | K-Means – fast.
Hierarchy – Accurate and slow

Matrix Factorization

It assumes that:

1) Each user can be described as k attributes or features: Eg: Feature 1 can be how (that says how close it is to sci-fi) movies
" 2 " " user liking sci-fi movies

2) Each item can be described by an analogous set of k attributes or features. Example, feature 1 would say on descale to some other feature.

3) If we multiply each feature of the user by the corresponding feature of the movie & add them all, this will be a good approximation for the rating of the user.

User Based R. Sys.

1) Each user is represented as a vector of u items
2) Cosine

UCF → Not great for more users & gave rating for 1 item then & if not watched then their vector is empty.

IBCF →

Not great for more items [when more movie coming up]

Great for old items [Since we can recommend similar item based on current item]

---

t-SNE: Best for data visualization, but takes time.

Evaluation for Recommender Systems.

- RMSE
- MAP : When we have binary feedback, it is best. It take account ranking or ordering
- NDCG : Normalize Discounted Cumulative Gain When feedback is numeric, it account position Ex: Enum or Ordinal

Implicit Feedback

- Explicit Feedback :

MPR & MRR

MAP (Mean average precision) : It must have true results at the top

NDCG

The relevance is discounted by

$\gamma_i = \frac{1}{\log_2(i+1)}$ and the sum @ k is normalized by its upper bound - the IDCG

$k = 3$

DCG (Discounted cumulative gain) @ k

$= \frac{1}{\log_2(1+1)} + 0 + \frac{5}{\log_2(3+1)}$ Rating = 3.5

Ranking for the trained model which is correct 3

Rating for Result 1

pos. no.

pos. no.

Ideal (for ideal ranking)

Rating obt

Rating of 2

Rating of 1

$IDCG@k = \frac{5}{\log_2(1+1)} + \frac{3}{\log_2(2+1)} + \frac{1}{\log_2(3+1)} = 7.39$

position nos

pos. nos

pos. nos.

$NDCG@k = \frac{3.5}{7.39} = 0.47$

If compared to position nos. 3, I am half way the ideal

Ground Truth

- True Result 1 : Rating 5
- True Result 2 : Rating 3
- True Result 3 : Rating 1

Rank Induced by algo

- True Result : 3
- -ve Result
- True Result 1
- True Result 2