Countplot

a. sns.countplot(df['survived'])

df['survived'].value_counts().plot(kind='bar')

b. Piechart

df['survived'].value_counts().plot(kind='pie', autopct='%.2f')

Numerical data

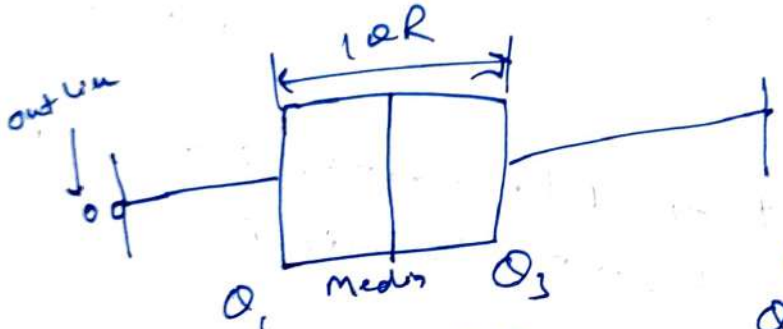
import matplotlib.pyplot as plt

plt.hist(df['Age'], bins=5)

we can change the bins value increase or decrease with respect to the bins. complexity of bars

Dist plot

sns.distplot(df['Age'])



min

$Q_1 - 1.5 * IQR$

Q_1 25 percentile

75 percentile

max

$Q_3 + 1.5 * IQR$

Box plot

sns.boxplot(df['fare'])

df['Age'].skew()

scatter plot

sns.scatterplot(tips['total_bill'], tips['tip'], hue=df['sex'])

Barplot

two (Numerical - categorical)

sns.barplot(df['Pclass'], df['Age'], hue=titanic['sex'])

titanic

titanic

Box plot (Numerical - categorical)

sns.boxplot(titanic['sex'], titanic['Age'], hue = titanic['survived'])

$x' = \frac{x - \lambda}{\lambda}$

Distplot (Numerical - categorical)

sns.distplot(titanic['Age'])

sns.distplot(titanic[titanic['survived']==0]['Age'], hist=False)

sns.distplot(titanic[titanic['survived']==1]['Age'], hist=False)

Heat map (Categorical - categorical)

pd.crosstab(titanic['Pclass'], titanic['survived'])

sns.heatmap(pd.crosstab(titanic['Pclass'], titanic['survived']))

titanic.groupby('Pclass').mean()['survived']*100

Cluster map Categorical - categorical

pd.crosstab(titanic['sibsp'], titanic['survived'])

sns.clustermap(pd.crosstab(titanic['sibsp'], titanic['survived']))

Pairplot iris.head()

sns.pairplot(iris)

sns.pairplot(iris, hue='species')

Lineplot (numerical - numerical)

Pandas profiling

! Pip install pandas-profiling

from pandas_profiling import ProfileReport

Prof = ProfileReport(df)

Prof.to_file(output_file='Output.html')

Feature Engineering is the process of using domain knowledge to extract feature from raw data. These features can be used to improve the performance of ML algorithm

Feature Engineering

Feature construction

feature selection

Feature extraction

feature handling

- Missing value imputation
- Handling categorical features
- Outlier detection
- feature scaling

feature scaling is a technique to standardize the independent features present in the data in a fixed range

feature scaling

Standardization

Normalization

- min max scaler
- Robust scaler

Also called Z score normalization

$$x' = \frac{x_i - \bar{x}}{\sigma}$$

when to use standardization

1. K-means
2. K-Nearest neighbours
3. PCA
4. ANN
5. Gradient descent

the is no need for standardization in Decision tree Random forest, Gradient boost algorithms

Normalization is a technique often applied as part of data prep. for machine learning. The goal of normalization is to change the values of numeric columns, in the dataset to use common scale, without distorting differences in the ranges of values or losing information.

- min max scaling
- mean normalization
- ~~max absolute scaling~~ max absolute scaling
- Robust scaling

Min max scaling

$$x'_i = \frac{x_i - x_{min}}{x_{max} - x_{min}}$$

we get it between 0 and 1

certificate of origin

invoice transform. transformer ColumnTransform

Pipelines chain together multiple steps so that the output of each step is used as input to the next step

- easy to apply the same preprocessing to train & test

Box-Cox transforms the exponent lex i.e a variable called λ that varies over the range of -5 to 5 and in the process of searching, we examine all values of λ finally we can choose the optimal value for your variable

$x > 0$

Yeo Johnson transform is somewhat an adjusted of Box-Cox transform, by which we can apply it to -ve numbers

Box-Cox

$$x_\lambda = \frac{x^\lambda - 1}{\lambda^\lambda} \qquad x_\lambda = \frac{x^\lambda - 1}{\lambda}$$

Yeo-Johnson

$$\psi(y, \lambda) = \begin{cases} \frac{y^\lambda - 1}{\lambda} & \lambda \neq 0 \\ \log y & \lambda = 0 \end{cases}$$

L 50 Linear Regression Campusx

Simple Linear regression model

completly linear / sort of linear — easy to learn

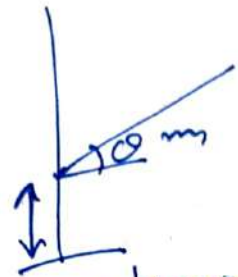
Line equation $y = mx + b$

How to find m and b

m = slope

b = y intercept

I'm sort of linear draw a line and called that line best fit line. i.e line which closely pass through all the points



x = cgpa

y = package

plt.scatter(df['cgpa'], df['package'])

x = df.iloc[:, 0:1]

y = df.iloc[:, -1]

lr.predict(x_test.iloc[0].values.reshape(1,1))

importance of b is that in case value of

m, b

dived

closed form solution

OLS ordinary least sq.

non closed form sol^n

(Gradient descent)

$$b = \bar{y} - m\bar{x}$$

$$m = \frac{\sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n} (x_i - \bar{x})^2}$$

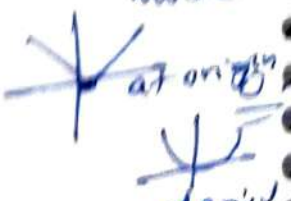
$\bar{x}$, $\bar{y}$ → mean

why modulus is not used instead of square

$E = d_1 + d_2 + d_3 + \dots + d_n$

$E = d_1^2 + d_2^2 + d_3^2 + \dots + d_n^2$

$E = \sum_{i=1}^{n} d_i^2$ ← Error function



derivation and origin must in modulus

take m, and b such that value of E is decreases

$d_i = (y_i - \hat{y}_i)$

$E = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$ Average error

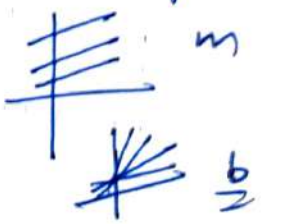
$E = \sum_{i=1}^{n} (\hat{y}_i - \bar{y}_i)^2$ total average

$\hat{y}_i = m x_i + b$

$E(m,b) = \sum_{i=1}^{b} (y_i - m x_i + b)^2$

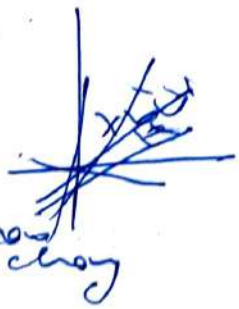
change this two

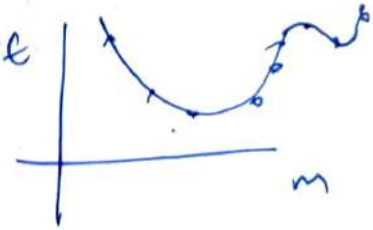
To find the value of m, and b which give minimum $E(m,b)$



$y = f(x)$

$E(m) = \sum_{i=1}^{m} (y_i - m x_i)^2$





$\frac{\partial E}{\partial b} = \frac{\partial}{\partial b} \sum_{i=1}^{n} (y_i - m x_i - b)^2 = 0$

$\sum \frac{\partial}{\partial b} (y_i - m x_i - b)^2 = 0$.

use chain rule

$$\sum -2(y_i - mx_i - b) = 0$$

divide both side by -2

$$\sum (y_i - mx_i - b) = 0$$

$$\sum y_i - \sum mx_i - \sum b = 0$$

divide both side by n

$$\frac{\sum y_i}{n} - \frac{\sum mx_i}{n} - \frac{\sum b}{n} = \frac{0}{n}$$

$$\bar{y} - m\bar{x} - \frac{nb}{n} = 0$$

$$\bar{y} - m\bar{x} = b$$

$$\boxed{b = \bar{y} - m\bar{x}}$$

$$E = \sum (y_i - mx_i - \bar{y} + m\bar{x})^2$$

$$\frac{dE}{dm} = \sum \frac{\partial}{\partial m} (y_i - mx_i - \bar{y} + m\bar{x})^2 = 0$$

$$\Rightarrow \sum 2(y_i - mx_i - \bar{y} + m\bar{x})(-x_i + \bar{x}) = 0$$

$$= \sum -2(y_i - mx_i - \bar{y} + m\bar{x})(x_i - \bar{x}) = 0$$

divide by -2

$$\sum (y_i - mx_i - \bar{y} + m\bar{x})(x_i - \bar{x}) = 0$$

$$\sum [(y_i - \bar{y}) - m(x_i - \bar{x})](x_i - \bar{x}) = 0$$

$$= \sum [(y_i - \bar{y})(x_i - \bar{x}) - m(x_i - \bar{x})^2] = 0$$

$$= \sum (y_i - \bar{y})(x_i - \bar{x}) = m \sum (x_i - \bar{x})^2$$

$$\boxed{m = \frac{\sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n} (x_i - \bar{x})^2}}$$

maximum minimum error function

$$\frac{dE}{dx} = 0$$

$$\frac{\partial E}{\partial m} = 0, \frac{\partial E}{\partial b} = 0$$

Regression Metrics

(1) MAE (2) MSE (3) RMSE (4) R2 S

(5) Adjusted R2 score

MAE Disadvantage: Modulation function is not differentiable at zero

$$MAE = \frac{\sum_{i=1}^{n} |y_i - \hat{y}_i|}{n}$$

Advantage:
1. Same unit
2. Robust to outlier

$RMSE = \sqrt{MSE}$

Disadv: Not differentiable

R2 score

$$1 - \frac{SSR}{SSM}$$

SSR - Sum of sq. error in Regression line

SSM: Sum of sq. error in mean line

Flaw: Regardless by adding irrelevant feature result improve

$$= 1 - \frac{\left[\sum_{i=1}^{n} (y_i - \hat{y}_i)^2\right]_{Reg}}{\left[\sum_{i=1}^{n} (y_i - \bar{y})^2\right]_{mean}}$$

Adjusted R2 sq

$$R^2 Adj = 1 - \left[\frac{(1-R^2)(n-1)}{(n-1-k)}\right]$$

$$MSE = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

Adv: loss function is differentiable

disadv: not in same unit as output

Multiple Linear regression

SLR — MLR — PLR / Polynomial

cgpa / gender / iq | LPA

x_1 | x_2 | x_3 | y cgpa | Lpa

just extension of linear regression

$y = mx + b$ ($m \to \beta_1$, $b \to \beta_0$) $\quad y = mx_1 + mx_2 + b$

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2$$

$$4D = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3$$

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_n x_n$$

$$y = \beta_0 + \sum_{i=1}^{n} \beta_i x_i$$

$n = 1$

$$y = \beta_0 + \beta_1 x_1$$

3D = $y = \beta_0 + \beta_1 x_1 + \beta_2 x_2$

x_1 = CGPA
x_2 = iq
y = LPA

$LPA = \beta_0 + \beta_1 \times cgpa + \beta_2 \times iq$

these coefficients are like weights

if β_2 is very small then value of iq became very small.

Lecture 54 Multiple Linear Regression Part 2

How find $\beta_0, \beta_1, \beta_2 \cdots \beta_n$?

eg.

cgpa	iq	gender	lpa
x_1	x_2	x_3	y actual

4D plots

$$\hat{y} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3$$

$\hat{y}$ Predicted

$y = mx + b$

$y = \beta_0 + \beta_1 x$

100 students

$$y_1 = \beta_0 + \beta_1 x_{11}$$

$$\hat{y} = \begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \vdots \\ \hat{y}_{100} \end{bmatrix} = \begin{bmatrix} \beta_0 & \beta_1 x_{11} & \beta_2 x_{12} & \beta_3 x_{13} \\ \beta_0 & \beta_1 x_{21} & \beta_2 x_{22} & \beta_3 x_{23} \\ \vdots & & & \\ \beta_0 & \beta_1 x_{100,1} & \beta_2 x_{100,2} & \beta_3 x_{100,3} \end{bmatrix}$$

assuming 100 rows $\rightarrow$ n rows 4 cols $\rightarrow$ 3 cols $\rightarrow$ m cols

$$\hat{y} = \begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \vdots \\ \hat{y}_n \end{bmatrix} = \begin{bmatrix} \beta_0 & \beta_1 x_{11} & \beta_2 x_{12} & \beta_3 x_{13} \cdots \beta_m x_{1m} \\ \beta_0 & \beta_1 x_{21} & \beta_2 x_{22} & \beta_3 x_{23} \cdots \beta_m x_{2m} \\ \vdots & & & \\ \beta_0 & \beta_1 x_{n1} & \beta_2 x_{n2} & \beta_3 x_{n3} \cdots \beta_m x_{nm} \end{bmatrix}$$

xwrite matrix

$$= \begin{bmatrix} x_{11} & x_{12} & x_{13} & \cdots & x_{1m} \\ x_{21} & x_{22} & x_{23} & \cdots & x_{2m} \\ \vdots & & & & \\ x_{n1} & x_{n2} & x_{n3} & \cdots & x_{nm} \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \vdots \\ \beta_m \end{bmatrix}$$

A B = A B

$$\hat{y} = X\beta$$

$$y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} \quad e = y - \hat{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} - \begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \vdots \\ \hat{y}_n \end{bmatrix}$$

$$e = \begin{bmatrix} y_1 - \hat{y}_1 \\ y_2 - \hat{y}_2 \\ \vdots \\ y_n - \hat{y}_n \end{bmatrix}$$

single LR

$$e = \sum_{i=1} (y_i - \hat{y}_i)^2$$

$$E = e^T e \qquad \begin{bmatrix} y_1 - \hat{y} \\ y_2 - \hat{y}_2 \\ \vdots \\ y_n - \hat{y}_n \end{bmatrix} = e$$

$$\begin{bmatrix} (y_1 - \hat{y}_1) & (y_2 - \hat{y}_2) & (y_3 - \hat{y}_3) \cdots (y_n - \hat{y}_n) \end{bmatrix}$$

$$\begin{bmatrix} y_1 - \hat{y}_1 \\ y_2 - \hat{y}_2 \\ y_3 - \hat{y}_3 \\ \vdots \\ y_n - \hat{y}_n \end{bmatrix}$$

$$(y_1 - \hat{y}_1)^2 + (y_2 - \hat{y}_2)^2 + (y_3 - \hat{y}_3)^2 \cdots + (y_n - \hat{y}_n)^2$$

$$\sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

$$E = e^T e = (y - \hat{y})^T (y - \hat{y})$$

$$= (y^T - \hat{y}^T)(y - \hat{y})$$

$$= [y^T - (X\beta)^T](y - X\beta)$$

$$E = y^T y - y^T X\beta - (X\beta)^T y + (X\beta)^T X\beta$$

$$(A + B)^T = A^T + B^T$$

$$(A - B)^T = A^T - B^T$$

Datum $y^T X\beta = (X\beta)^T y$ Proof

$$y = A \quad X\beta = B$$

$$A^T B = B^T A$$

$(AB)^T = B^T A^T$, $(A+B)^T = B^T + A^T$

$A^T B = B^T A$ — (1) To be proved

$A^T B = (A^T B)^T \rightarrow A^T B = C$ $\boxed{C = C^T}$

$(Y^T X \beta)^T = Y^T X \beta$

$Y = \begin{bmatrix} \\ \\ \\ \end{bmatrix}$ $Y^T = (1 \times n)$

~~$1 \times (m+1)$~~

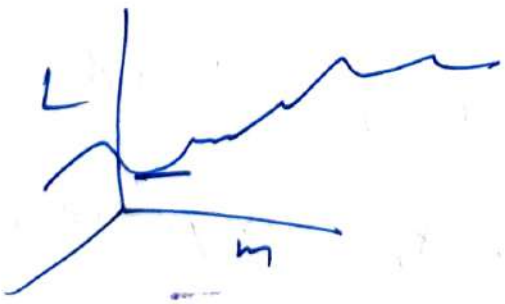
$m \times 1 \rightarrow (1 \times n)\ (n \times (m+1))\ [(m+1) \times 1]$

$1 \times (m+1)$ $(m+1) \times 1$

$1 \times 1 = [7]^T = [7]$

$\longrightarrow$

$E = Y^T Y - 2 Y^T X \beta + \beta^T X^T X \beta$



$\frac{dE}{d\beta} = \frac{d}{d\beta}\left[Y^T Y - 2 Y^T X \beta + \beta^T X^T X \beta \right] = 0$

$Y^T Y$ is constant, so its derivative is 0

$= 0 - 2y^T x + \frac{d}{d\beta}\left[\beta^T x^T x \beta\right] = 0$ (sol)

matrix differentiation

$y = A^T x A \quad \frac{dy}{dA} = 2xA^T$

$= 0 = -2y^T x + 2x^T x \beta^T = 0$

$= 2x^T x \beta^T = 2y^T x$

$\beta^T = \frac{A}{x^T x} \quad \beta^T = y^T x (x^T x)^{-1}$

$(\beta^T)' = [y^T x (x^T x)^{-1}]^T$

$\beta = [(x^T x)^{-1}]^T (y^T x)^T$

$\beta = [(x^T x)^{-1}]^T x^T y$

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

$x = n \times (m+1)$

$(m+1) \times n \quad n \times (m+1)$

$(m+1) \times (m+1)$

$(AB)^T = B^T A^T \quad , (A^T B)^T = B^T A$

$A^T B = B^T A$ to be proved

$A^T B = (A^T B)^T$

$\rightarrow A^T B = C$

$C^T = C$

$(y^T x \beta)^T = y^T x \beta$

$b = n \times (m+1)$

$m \times 1 \rightarrow (1 \times m)(n \times (m+1)) [(m+1) \times 1]$

$[1 \times 1] \quad [1 \times (m+1)] \quad [(m+1) \times 1]$

Agrohi L-13 Vanishing gradient

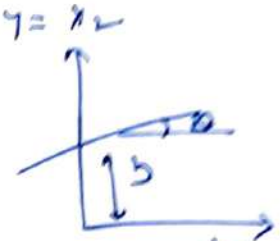
Svm complex — Margin maximizing hyperplane

map ⓐ

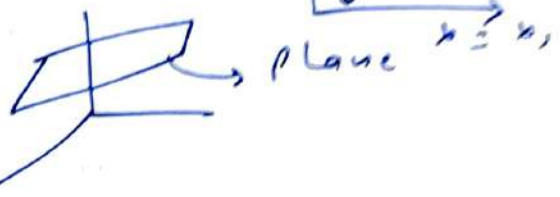
$w^T x + b$

1. very Robust to outliers
2. It can also work on nonLinear data by using Kernel
3. Applicable for both classification and Regression problem

Equation of a Hyperplane



4D, 5D, nD Hyperplane



for 2D :- $y = mx + b$

↓ general form

$ax + by + c = 0$

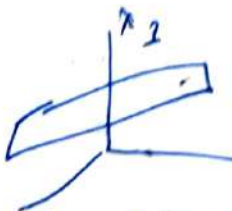
$y = -\frac{a}{b}x - \frac{c}{b}$

$m = -\frac{a}{b}$, $b = -\frac{c}{b}$

$ax_1 + bx_2 + c = 0$

$w_1 x_1 + w_2 x_2 + w_0 = 0$

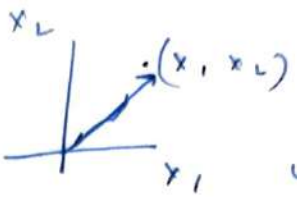
for 3D



→ $w_1 x_1 + w_2 x_2 + w_3 x_3 + w_0 = 0$

for Hyperplane n dimension :-

$w_1 x_1 + w_2 x_2 + w_3 x_3 - - - w_n x_n + w_0 = 0$



$w_1 x_1 + w_2 x_2 \pm w_0 = 0$

$[w_1, w_2]$ $w \cdot x = w_1 x_1 + w_2$

$w = [w_1, w_2, \cdots w_n]$

$x = [x_1, x_2 \cdots x_n]$

$w \cdot x + w_0 = 0$

$$\begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix} \quad \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

$$[w_1\ w_2\ w_3 \cdots w_n] \quad \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix}$$

$$w^T x + w_0 = 0$$

$$\therefore a \cdot b = a^T b$$

$w_1 x + w_2 y + w_0 = 0$

$b = -\frac{w_0}{w_2}$

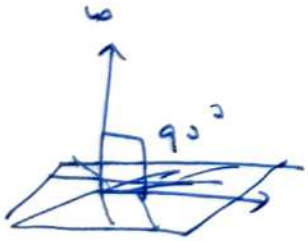
$w^T x = 0$

$w_1 x_1 + w_2 x_2 = 0$

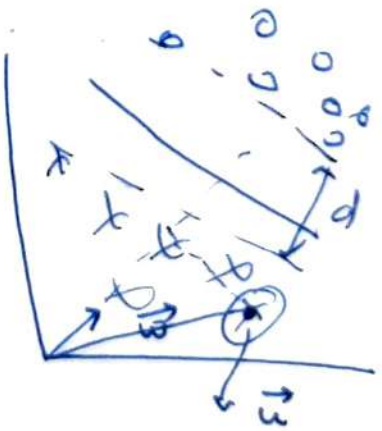
$w_1 x_1 + w_2 x_2 \cdots w_n x_n = 0$

$w^T x = w \cdot x = \|w\| \|x\| \cos\theta = 0$

when $\theta = 90^\circ$



Mathematics for SVM



Decision Rule

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

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

$\vec{w} \cdot \vec{u} + b \geq 0$

+ve / -ve point

$2x + 3y + 1 = 0$

$w_1 x_1 + w_2 x_2 + w_0 = 0$

$\vec{w} = (2, 3) \quad \vec{u} = (3, 2)$

$\vec{w} = (2, 3)$

$2(3) + 3(2) + 1 \geq 0$

$\vec{w} = (-2, 0)$

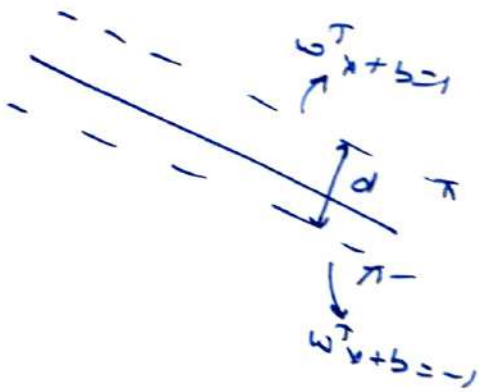
$2(-2) + 3(0) + 1$

$-1 \leq 0$

for any $\vec{x}_i$

$$\hat{y} = \begin{cases} +1 & \text{if } \vec{w} \cdot \vec{x}_i + b \geq 0 \\ -1 & \text{if } \vec{w} \cdot \vec{x}_i + b < 0 \end{cases}$$

1. why?
2. why equal
3. Some other number



constraints

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

for support vector

$y_i (\vec{w} \cdot \vec{x}_i + b) = 1$

$d = (\vec{x}_2 - \vec{x}_1) \cdot \frac{\vec{w}}{\|w\|}$

$= \frac{\vec{x}_2 \vec{w} - \vec{x}_1 \vec{w}}{\|w\|}$

$y_i (\vec{w} \cdot \vec{x}_i + b) = 1$

$1 (\vec{w} \vec{x}_i + b) = 1.$

$\vec{w} \cdot \vec{x}_i = (1-b)$ ←

$-1(\vec{w} \cdot \vec{x}_i + b) = 1$

$\vec{w} \cdot \vec{x}_i = -b-1$

$$\frac{(1-b)-(b-1)}{\|w\|}$$

$$= \frac{1-b+b+1}{\|w\|}$$

$$d = \frac{2}{\|w\|}$$

$$\underset{(w^*, b^*)}{\text{argmax}} \frac{2}{\|w\|} \quad \text{such that} \quad y_i(\vec{w} \cdot \vec{x}_i + b) \geq 1$$

Hard margin SVM

HARdHoRN

Soft Margin SVM

$$\underset{(w^*, b^*)}{\text{argmin}} \frac{\|w\|^2}{2} + C \sum_{i=1}^{n} \xi_i$$

ξ → zeta

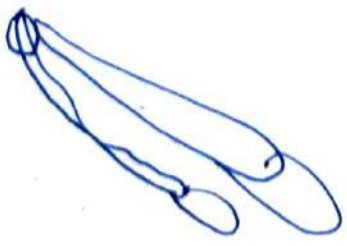
ξ_i → hinge loss

Margin error + classification error

Logistic loss + $\lambda \|w\|$

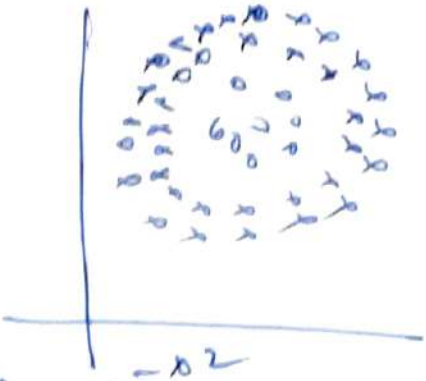
$\xi_i = 0$ for correctly classified pt.

$C \propto \frac{1}{\lambda}$

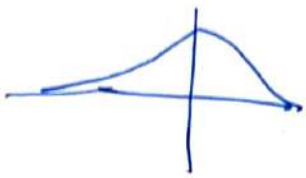


Kernel tricks sum

- RBF
- Polynomial
- Sigmoid



$f(x) = e^{-x^2}$

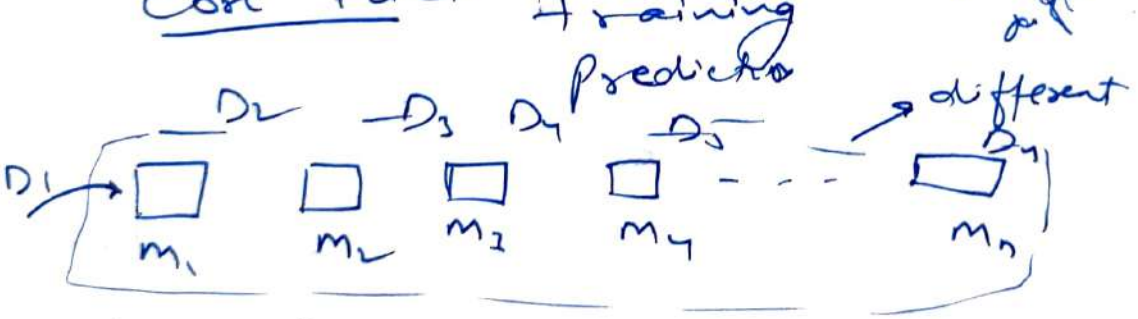


Gridsearchcv

Ensemble learning

Wisdom of crowd

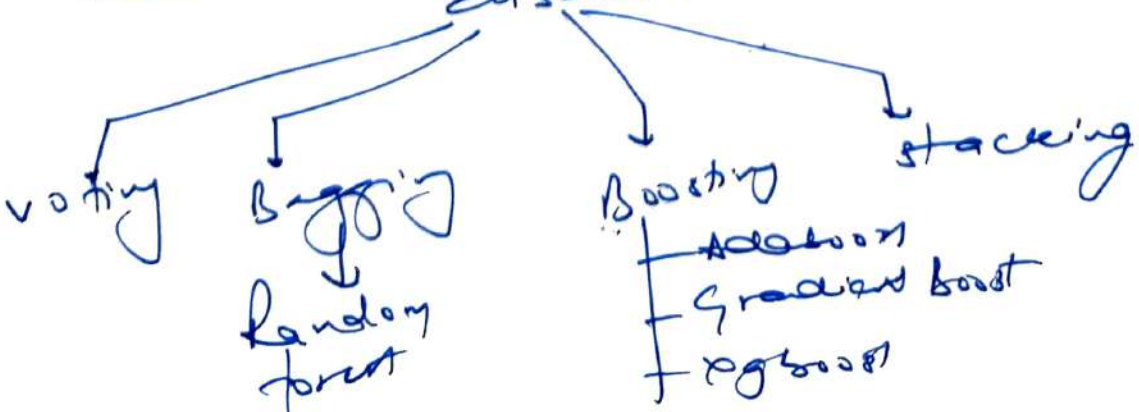
Core idea



same algo same data
diff algo diff data
same algo diff data

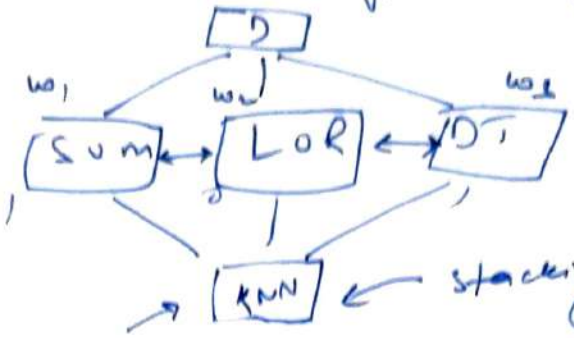
LR SVM DT

Types of Ensemble learning



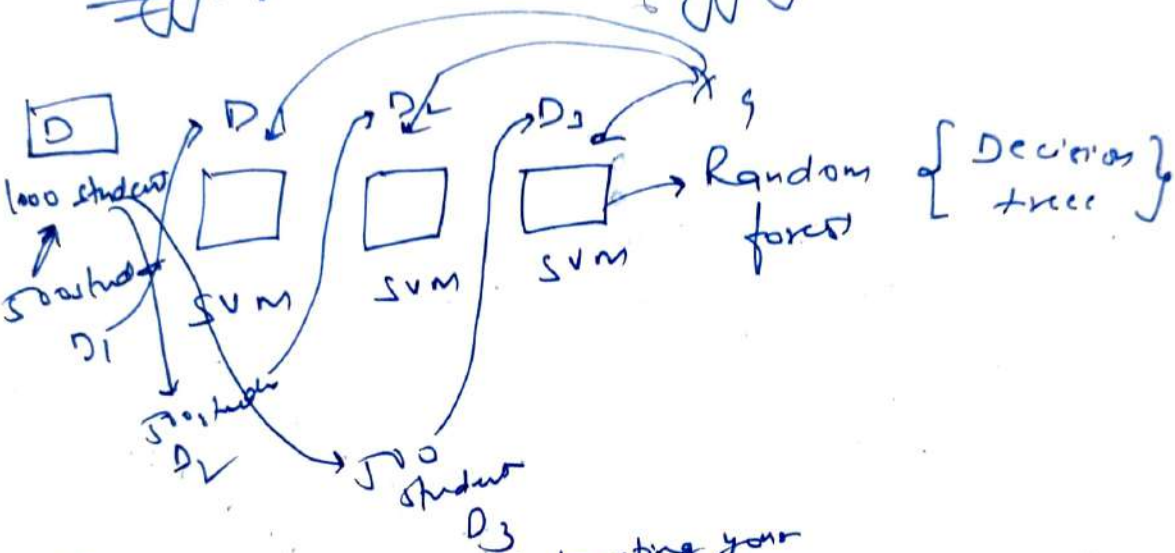
Voting

→ Base model are different algorithms

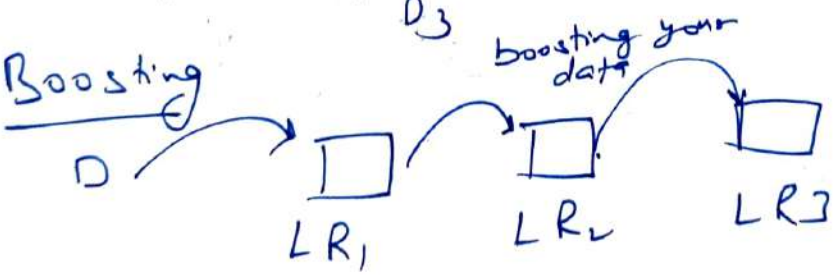


{ x_q }

Bagging → Bootstrap aggregation



Boosting



- computation cost expensive

Adv. 1. Improvement in performance

2. Bias - Variance reduce

3. Robustness.

Low Bias + low variance

(4) When to use :- Always use ensemble learning

When to use
- Always

Voting Ensemble

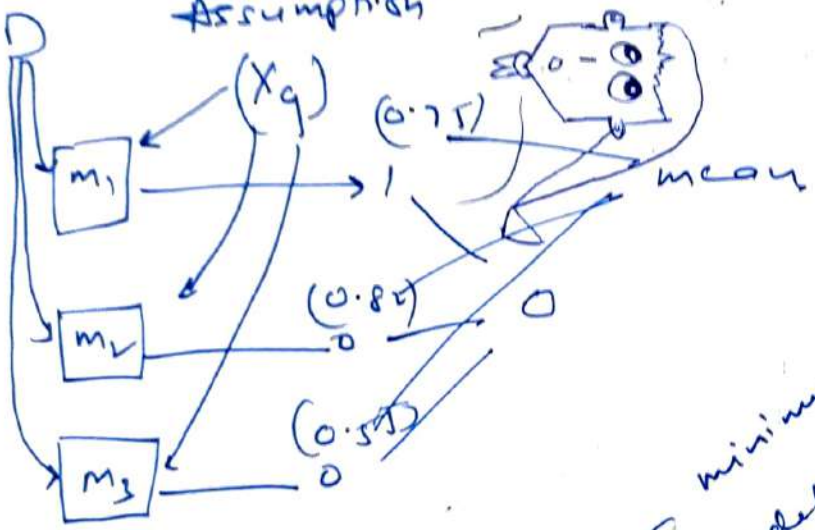
- Core idea / intuition
- Voting classifier
- Voting Regressor

Part 1 - Introduction

Core idea / intuition

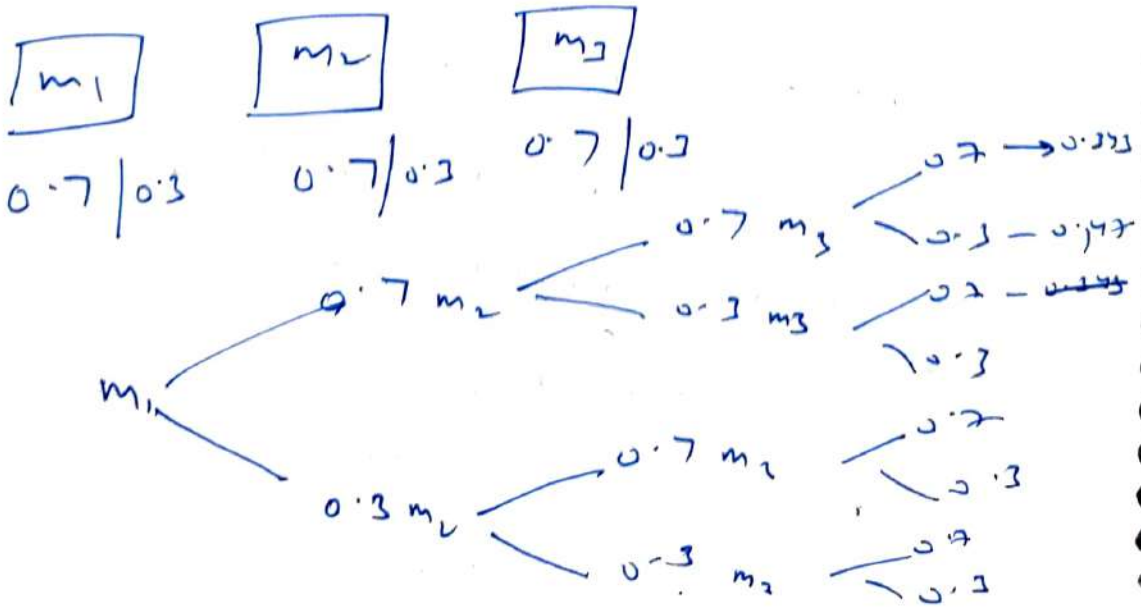
why voting works?

Assumption



- minimum accuracy 50%
- model independent

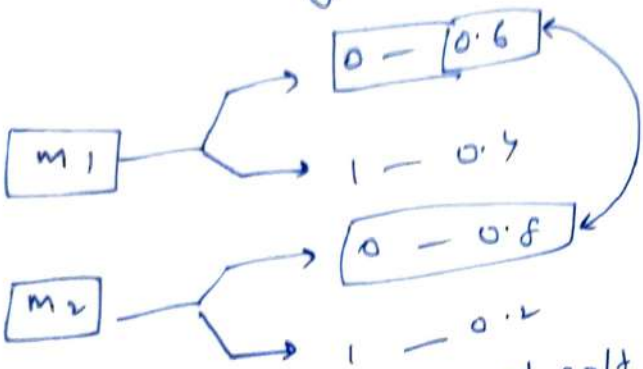
why voting works?



Classifier

- core idea / Demo
- Hard voting Vs soft voting

- code
- Hyperparameters



0 - 0.7
1 - 0.3

Average calculation

* Run both hard and soft voting and select one which is better.

Part 3 - Voting Regressor

- core idea + Demo
- code
- Hyperparameters

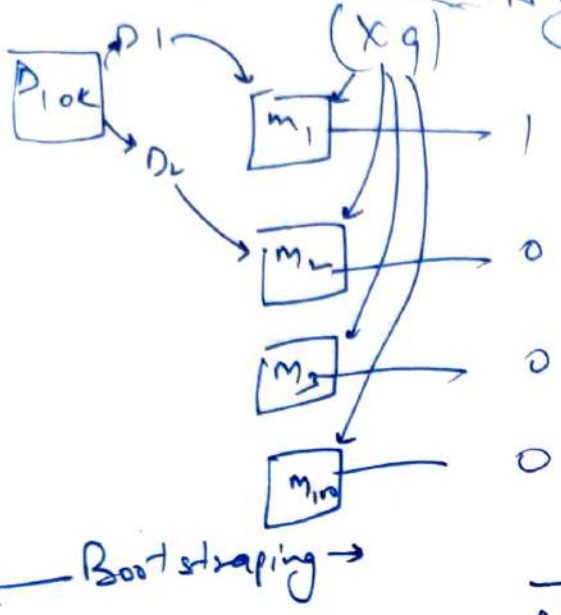
Bagging Ensemble learning

Part 1

core idea /
why use Bagging
when to use Bagging
code demo
Types of Bagging

Bagging → Bootstrap
→ Aggregation

+ Same data different algorithm
- different data same algorithm



Majority vote
= 0

change data with different subset either with replacement or without replacement

Low bias Low variance
L B L V

L B H V
H V L B

Bias Variance

consistent result

when to use? Everytime
- Random forest
- DT - easy to understand

types:
1. Pasting (without replacement Row sampling)
2. Random subspace (do column sampling with or without replacement)
3. Random patches (both column and row replacement)

Part 2 Bagging classifier

63% in (37%) rows are not used

OOB score (out of Bagging) — local / imp

Part 3 Regressor

- Bagging generally give better results than pasting

Bagging regressor D → □ □ □ □ → mean

Random forest

- Decision tree, bagging technique

Random forest ↳ grp. of trees

Bagging → Bootstrapped Aggregation ↳ sampling (Random)

Random forest is like Bagging but only difference is b/w in RF we use Decision tree

D = 1000 rows →

DT1 DT2 DT3 ... DT100

- Row sampling
- Column sampling
- combination

with replacement
without replacement

Bootstrap

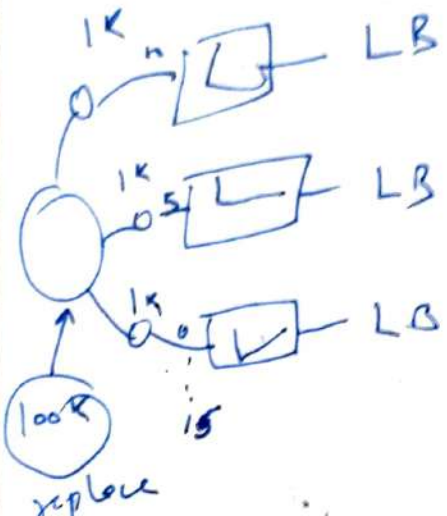
Aggregation (Most frequent)

- How Random forest perform so well? Bias variance trade off

[LB LV]

$B \propto \frac{1}{V}$.

Random forest [LB tree] connected to [LB LV]



Random forest Hyperparameters

sklearn.ensemble.RandomForestClassifier

max_sampling
max_features

(Complex Decision tree hyperparameters)

Hpt Hyperparameter Tuning Rf using GridSearchCV and RandomizedSearchCV

OOB Evaluation — out of Bag

Sampling with replacement.

out of bag sample :- which are not selected in model training

OOB hidden → testing → Performance

BTW, rows OOB sample used validation sample

feature Importance using Rf and DT. How feature importance calculated?
feature column importance?

Feature importance

$$n_i = \frac{N-t}{N}\left[\text{impurity} - \left(\frac{N-t-r}{N-t} \times \text{right\_impurity}\right) - \left(\frac{N-t-l}{N-t} \times \text{left\_impurity}\right)\right]$$

$$f'_k = \frac{\sum_{j \in \text{node split on feature } k} n_j}{\sum_{j \in \text{all nodes}} n_j}$$

How Ada boost classifier works?

1. weak learners :- just over 50% Accuracy
2. Decision stumps : max depth 1
3. +1 and -1

stage wise, additive method

$$A = \text{Sign}(\alpha_1 h_1(x) + \alpha_2 h_2(x) + \alpha_3 h_3(x))$$

Ada Boost - step by step

$$\alpha = \frac{1}{2}\ln\left(\frac{1-\text{error}}{\text{error}}\right)$$

$$\alpha_1 = \frac{1}{2}\ln\left(\frac{1-0.4}{0.4}\right) = 0.20$$

↑ wt of mis classified pt
↓ wt of correct classified pt

For misclassified

new_wt = curr_wt $\times e^{\alpha_1}$

For correctly classified

new_wt = curr_wt $\times e^{-\alpha_1}$

normalized the updated weights

x_1 x_2 y new_wt - range

5 random no. b/w 0 and 1

0.13
0.43
0.62
0.50
0.8

Bagging

Bagging Vs Boosting

1) Type of model used
2) Sequential vs Parallel
3) weightage of base learners

df.duplicated().sum() ① df.shape ② df.describe()
③ df.info(), ④ df.sample(5), ⑤ df.isnull().sum(),
⑥ df.duplicated().sum() ⑦ df.corr()['Survived']

Univariate analysis ① Categorical Data:

① Countplot
sns.countplot(df['Embarked'])

② Piechart
df['sex'].value_count().plot(kind='pie', autopct='%.2f')

② Numerical data:

① Histogram:- import matplotlib.pyplot as plt
plt.hist(df['Age'], bins=5)

② Distplot sns.distplot(df['Age'])

③ Boxplot sns.boxplot(df['Age'])

df['Age'].min() ; df['Age'].max()
df['Age'].mean(), df['Age'].skew()

Bivariate analysis scatterplot (Numerical - Numerical)

sns.scatterplot(tips['total_bill'], tips['tip'],
hue=df['sex'], style=df['smoker'], size=df(size))

Barplot (Numerical - categorical)

sns.barplot(titanic['Pclass'], titanic['Age'],
hue = titanic['sex'])

3. Box plot (Numerical - categorical)

sns.boxplot(titanic['sex'], titanic['Age'],
hue = titanic['survived'])

4. Distplot (Numerical - categorical)

sns.distplot(titanic[titanic['survived']==0]['Age'],
hist = False)

sns.distplot(titanic[titanic['survived']==1]['Age'],
hist = False)

5. Heat map (categorical - categorical)

sns.heatmap(pd.crosstab(titanic['Pclass'],
titanic['survived']))

(titanic.groupby('Embarked').mean()['survived']*100)

6. Cluster Map (categorical - categorical)

sns.clustermap(pd.crosstab(titanic['Parch'],
titanic['Survived']))

7. Pair Plot sns.pairplot(iris, hue = 'species')

8. Line plot (Numerical - Numerical)

new = flights.groupby('year').sum().
reset_index()

sns.lineplot(new['year'], new['passengers'])

sns.clustermap(flights.pivot_table(values = 'passengers',
index = 'month', columns = 'years'))

```
from sklearn.preprocessing import OrdinalEncoder
oe = OrdinalEncoder(categories=[['Poor','Average','Good'],
                                ['school','UG','PG']])
X_train = oe.transform(X_train)
```

(19) One Hot Encoding using pandas

```
pd.get_dummies(df, columns=['fuel','owner'])
```

(20) k-1 One Hot encoder

```
pd.get_dummies(df, columns=['fuel','owner'],
               drop_first=True)
```

header=1 colspecs, squeeze=True, skiprows=[0,1]

nrows=100, Encoding parameter, encoding='Latin-1'

skip bad line:- —, sep=',', encoding='Latin-1',

dtype parameter error_bad_line=False

, dtype {'target':int})

Handling Dates

Converters : parse_date=['date'])

```
def rename(name):
    if name == "Royal challenger":
        return RCB
    else:
        return name
converter = {'team1':rename}
```

15 na_value Parameter
, na_values = ['Male']

16. Loading a huge dataset in chunks
, chunksize = 5000)

for chunks in df1:
print(chunk.shape)

Json / sQL

Java script on notation pd.read_json(' ')

working with sQL
- Download file → Xampp control panel

! pip install mysql connector
import mysql.connector

How big data is
How does the data look like?
what is the data type of cols?
Are there all missing values
How does the data looks mathematically
Are there duplicate values?
df.duplicated().sum()

How is the correlation b/w cols
df.corr()['survived']
Univariant Analysis

, hue = df['sex'], style = df['smoker'], size = df[size]

pd.crosstab(titanic['Pclass'], titanic['survived'])

sns.pairplot(iris, hue='species')

Feature Engineering (extract features from raw data)

- Feature transformation
  - Missing values
  - Handling categorical values
  - Outlier detection
  - Feature scaling
- Feature construction
- Feature selection
- Feature extraction

8837850

Feature scaling to standardize the independent feature present in the data in a fixed range

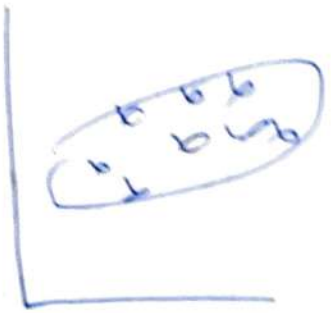
Similar range eg 10 - 100

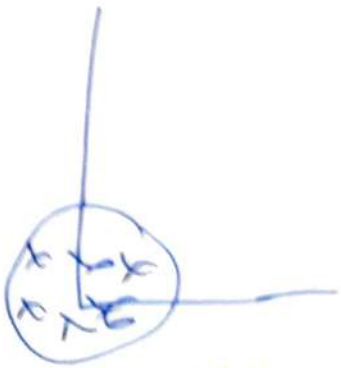
0.1 0.001

Feature scaling

- standardization (Z score normalization)
- Normalization
  - minmax scaler
  - Robust scaler

$$x_i' = \frac{x_i - \bar{x}}{\sigma}$$





① K means ② K nearest neighbours ③ PCA
④ ANN ⑤ Gradient Descent

Normalization Min max scaling

is technique often applied as part of data prep. for ML. The goal of normalization is to change the values of the numeric columns in the dataset to use a common scale, without distorting differences in the ranges of values or losing information

weight
130
67
81
61
22
57
⋮

- Min Max scaling
- mean normalization
- max absolute
- Robust scaling

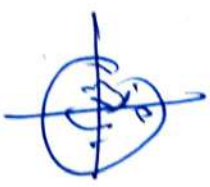
Normalization min max scaling

mean centering

$$x_i = \frac{x_i - x_{mean}}{x_{max} - x_{min}}$$

range [0, 1]

$$x_i = \frac{x_i - x_{mean}}{x_{max} - x_{min}}$$



Max abs scaling $x' = \frac{x_i}{|x_{max}|}$

Robust scaling $x_i' = x_i - x_{median}$ / IQR [ ]

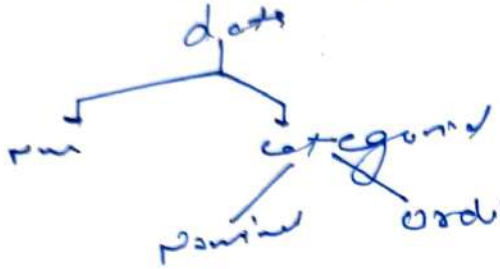
Normalization Vs standardization

1)

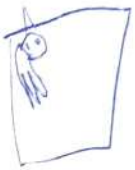
+ Robust scaling

Ordinal Encoding | Label encoding

Encoding categorical variables



Label encoder use for only target values not for input values



One Hot Encoding

Dummy variable trap

use (n-1) instead of (n)

pd.get_dummies(df, columns=[ ], drop_first=True)

Pipelines chains together multiple steps so that the output of each step is used as input to the next step

Pipelines make it easy to apply the same preprocessing to train and test

- log transform
- Reciprocal
- power (sq / sqrt)

- i) Box-cox
- ii) Yeo-Johnson

- normal distribution

sklearn

- function transfor
- Power transfor
- Quantile transformer

sns.displot pd.skew() = 0

QQplot

Box cox transform

$$x_i(\lambda) = \begin{cases} \dfrac{x_i^{\lambda} - 1}{\lambda} & \text{if } \lambda \neq 0 \\ \ln(x_i) & \text{if } \lambda = 0 \end{cases}$$

$\lambda < 0$

The exponent here is a variable called (λ) that varies over the range of -5 to 5, and the process of searching, we examine all values of λ. Finally we choose the optimal value (resulting in the best approximation to the normal distribution) for your variable

Yeo-Johnson transform

$$x_i'(\lambda) = \begin{cases} [(x_i+1)^{\lambda} - 1]\,\lambda \\ \ln(x_i) + 1 \\ -[(-x_i+1)^{2-\lambda}] \end{cases}$$

log :- big data , Right skewed data

Reciprocal $\frac{1}{x}$	$sq(x^2)$	sqrt $\sqrt{x}$	log
	Left skewed		Right sqrt skewed
	skewed		

$$x_i = \begin{cases} \frac{x_i^{\lambda} - 1}{\lambda} & \text{if } \lambda \neq 0 \\ \ln(x_i) & \text{if } \lambda = 0 \end{cases}$$

The exponent here is a variable called Lambda (λ) that varies over the range -5 to 5, and in the process of searching we examine all values of λ. Finally we choose the optimal λ value for your variable

good

Discretization :- is the process of transforming continuous variables into discrete variables by creating a set of contiguous intervals that span the range of the variable values. Discretization is also binning, when bin is an alternative name for interval

$\frac{\text{max} - \text{min}}{\text{bins}}$ $\quad \frac{100 - 0}{10} = 10$

0-10 , 10-20 , 20-30 -- 90-100

equal width bins

1) outlier
2) spread of data

(5) equal frequency / quantile Binning

0-16 16-20 20-22 , 22-25 - -

— Outlier
— value spread uniform

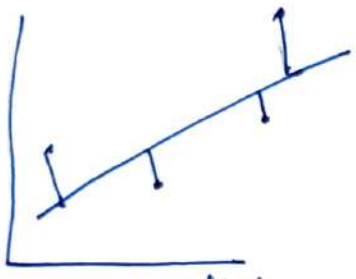
(6) K means Binning clustering

Gradient descent Complex in Brief

- Linear Regression
- Logistic Regression
- Deep learning ...

2 columns 4 Rows

cgpa	LPA
=	=
=	=
=	=
=	=



To find best fit line

$$L = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

$$\hat{y}_i = mx_i + b$$

$$L = \sum_{i=1}^{4} (y_i - mx_i - b)^2$$

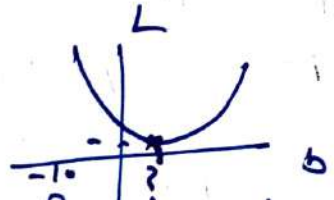
$L(m, b)$ depend on m and b

$m = 78.35$ (we know)

$$L = \sum_{i=1}^{4} (y_i - 78.35x_i - b)^2$$

$\min$

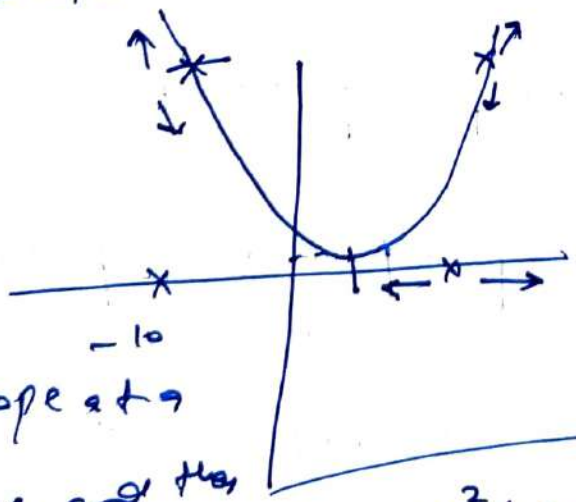
$L \to b^2$



OLS

Step 1 := select a Random b

Let $b = -10$



* find out the slope at the particular point.
by derivative and then put value of x.

$y = b^2 + 2$

$\frac{dy}{dx} = 2x = 2(5)$

$= 10$

- If slope is -ve b increased
If slope is +ve b decreased

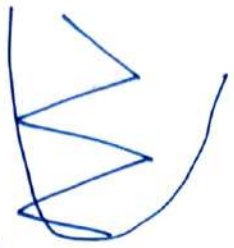
$b_{new} = b_{old} - slope$

η = learning rate

$b_{new} = b_{old} - \eta \, slope$

$\eta = 0.01, 0.1, 0.001, 0.0001$

If η is high or 1 then this type of change observed.



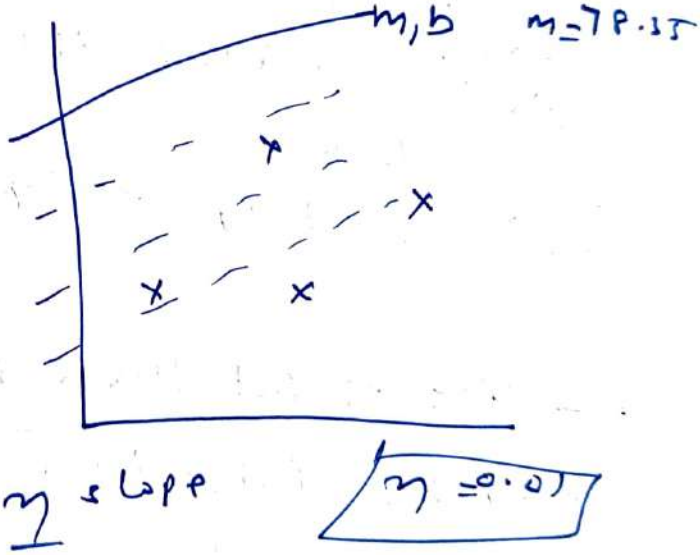
When to stop :-

$b_{new} - b_{old} = 0.0001$

1) > 0.0001

2) Iteration = epochs = 1000, 100

Mathematical formulation



Step 1 :- is always start with random value of $b_0 = b$

for i in epochs → 1000;

$b_{new} = b_{old} - \eta \, slope$

$\eta = 0.01$

$L = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$ Find slope $b = b$?

$b = 0$

$\frac{dL}{db} = \frac{d}{db} \left(\sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \right) =$

$\frac{d}{db} \sum_{i=1}^{n} (y_i - mx_i - b)^2 = -2 \sum_{i=1}^{n} (y_i - mx_i - b)$

$b = 0$

$$= -2 \sum_{i=1}^{n} (y_i - 78.35 * x_i - 0)$$

slope(b=0)

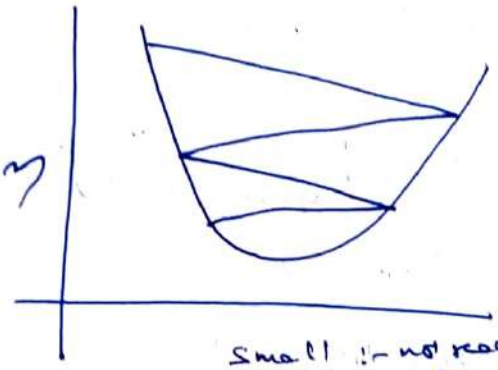
$b_{new} = b_{old} - \eta \, slope_{b=old}$

b_{new} i=1

Visualization

↓ learning rate η

↑ epochs.



small :- not reach the end

-large :- by pass the convergence

Few discussion

① Learning rate

② The universality of Gradient descend

$b = 0$

$b = b_{old} - \eta \, slope$ $\quad \frac{dL}{db} = [\quad]$

$\sum (y_i - \hat{y}_i)^2$ LR

LOR → L function

Adding m into the mix

step 1 :- initialise random value for m and b

$m = 1$

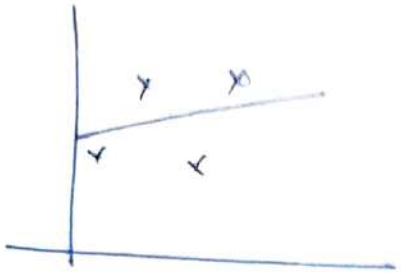
$b = 0$

step 2 Epoch = 10, lr = 0.01

for i in epochs:

$b = b - \eta \, slope$

$m = m - \eta \, slope$



$$L = \sum (y_i - \hat{y}_i)^2$$

$$L = \sum (y_i - mx_i - b)^2$$

$L(m, b)$

$z = f(x, y)$

$\frac{\partial L}{\partial b}$ $\frac{\partial L}{\partial m}$

$$\sum (y_i - mx_i - b)^2$$

$$\frac{\partial L}{\partial b} = \sum (y_i - mx_i - b)^2$$

$0 \quad 0 \quad -1$

$$= -2 \sum (y_i - mx_i - b)$$

$= \text{slope\_b at } b = 0$

Properly select the dir

$$\frac{\partial L}{\partial m} = -2 \sum (y_i - mx_i - b) x_i$$

slope_m at $m = 1$

Type of Gradient descent

1. Batch
2. stochastic
3. mini batch.

$m = 1, \quad b = 0$

$\frac{dL}{dm} \quad \frac{dL}{db}$

$$m_n = m_0 - \eta \times (\text{slope})_{m=0}$$

$$b_n = b_0 - \eta \times (\text{slope})_{b=0}$$

- use whole data, eg 300 row monitoring and then give decision
- stochastic:- only on single row, fast, error prone
- Minibatch in b/w batch and stochastic
  Use predefined batch size

cgpa | LPa (m, b)

cgpa / iq / gender / LPs

$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3$

Mathematical formulation

n-dim dataset 3-columns

cgpa	iq	lps
8.1	93	12
7.5	95	10.5

(2, 3)

$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2$

lpa (cgpa) (iq)

① Random values
$\beta_0 = 0$, $\beta_1, \beta_2 = 1$.

② epochs = 100, $lr = 0.1$

$\beta_0 = \beta_0 - \eta \cdot slope$

$\beta_1 = \beta_1 - \eta \cdot slope$

$\beta_2 = \beta_2 - \eta \cdot slope$

$L(\beta_0, \beta_1, \beta_2)$

four dimensional graph

$\frac{\partial L}{\partial \beta_0}$ $\frac{\partial L}{\partial \beta_1}$ $\frac{\partial L}{\partial \beta_2}$

n-dimension

$(n+1)$ $\beta_0 - \beta_3$

MSE

$$L = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)$$ — {rows 2, cols 2}

$$\frac{1}{2} \left[(y_1 - \hat{y}_1)^2 + (y_2 - \hat{y}_2)^2 \right.$$

no. of epochs → fixed

e = 100

Batch → 100

stochastic → 1000 → no. of rows

- Lack of confidence
- inconsistency

convex - global minima
- no local minima
non convex - local minima

when to use SGD?

① → Big data → SGD

② Non convex function.

Learning schedule:

Vary learning rate (lr) with epochs

SGDRegressor

Mini Batch GD (How frequently you update)

Batch :- group of rows

n = 1000

↳ batches → 100 → 10 batches

batches update per epoch

Ridge Regression

$$L = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 + \lambda m^2$$

overfitting where ML model in bottom training but on testing do not perform

m = slope

gather minimum this

$$L = \sum_{i=1}^{n} (y_i - mx_i - b)^2 + \lambda m^2 \rightarrow \frac{\partial L}{\partial b} = 0 \rightarrow b$$

$b = \bar{y} - m\bar{x}$

$\bar{y}$ = y-mean

$\bar{x}$ = x-mean

m is slope

$\frac{\partial L}{\partial m} = 0 \rightarrow m$

$$L = \sum_{i=1}^{n} (y_i - mx_i - \bar{y} + m\bar{x})^2 + \lambda m^2$$

$$\frac{\partial L}{\partial m} = 2\sum_{i=1}^{n} y_i - mx_i - \bar{y} + m\bar{x})(-x_i - \bar{x}) + 2\lambda m = 0$$

$$= -\sum_{i=1}^{n} (y_i - \bar{y} - mx_i + m\bar{x})(x_i - \bar{x}) + \lambda m = 0$$

$$= \lambda m - \sum_{i=1}^{n} (y_i - \bar{y}) - m(x_i - \bar{x})](x_i - \bar{x}) = 0$$

$$= \lambda m - \sum_{i=1}^{n} (y_i - \bar{y})(x_i - \bar{x}) - m(x_i - \bar{x})^2 = 0$$

$$= \lambda m = \sum_{i=1}^{n} (y_i - \bar{y})(x_i - \bar{x}) + m\sum_{i=1}^{n} (x_i - \bar{x})^2 = 0$$

$$= \lambda m + m\sum_{i=1}^{n} (x_i - \bar{x})^2 = \sum_{i=1}^{n} (y_i - \bar{y})(x_i - \bar{x})$$

$$= m = \frac{\sum_{i=1}^{n} (y_i - \bar{y})(x_i - \bar{x})}{\sum_{i=1}^{n} (x_i - \bar{x})^2 + \lambda}$$

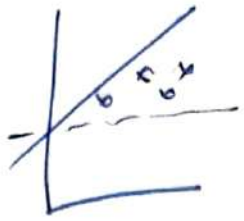
$\lambda \rightarrow$ hyperparameter alpha

$$= m = \frac{\sum_{i=1}^{n} (y_i - \bar{y})(x_i - \bar{x})}{\sum_{i=1}^{n} (x_i - \bar{x})^2} \qquad \lambda =$$

$y = mx + b$

reduce m value to get it on right value

bias ↑ variance ↓



$$L = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 + \lambda m^2$$

$$L = \sum_{i=1}^{n} y_i - mx_i - b)^2 + \lambda m^2$$

m = slope

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

$\frac{\partial L}{\partial m} = 0 \rightarrow m$

$b = \bar{y} - m\bar{x}$ $\quad \bar{y} \rightarrow y\text{-mean}$ $\quad \bar{x} \rightarrow x\text{-mean}$

m = slope

$$L = \sum_{i=1}^{n} (y_i - mx_i + \bar{y} + m\bar{x})^2 + \lambda m^2$$

$$\frac{\partial L}{\partial m} = 2 \sum_{i=1}^{n} (y_i - mx_i - \bar{y} + m\bar{x})(-x_i + \bar{x}) + 2\lambda m = 0$$

$$= -2 \sum_{i=1}^{n} (y_i - \bar{y} - mx_i + m\bar{x})(x_i - \bar{x}) + 2\lambda m = 0$$

$$= \lambda m = \sum_{i=1}^{n} [(y_i - \bar{y}) - m(x_i - \bar{x})](x_i - \bar{x}) = 0$$

$$= \lambda m = \sum_{i=1}^{n} (y_i - \bar{y})(x_i - \bar{x}) - m(x_i - \bar{x})^2 = 0$$

$$= \lambda m - \sum_{i=1}^{n} (y_i - \bar{y})(x_i - \bar{x}) + m \sum_{i=1}^{n} (x_i - \bar{x})^2 = 0$$

↑ ↑

$= \lambda_{n=m}$

$= \lambda_n + n \sum_{i=1}^{n} (x_i - \bar{x})^2 = \sum_{i=1}^{n} (y_i - \bar{y})(x_i - \bar{x})$

$$= m = \frac{\sum_{i=1}^{n} (y_i - \bar{y})(x_i - \bar{x})}{\sum_{i=1}^{n} (x_i - \bar{x})^2 + \lambda}$$

$\lambda \rightarrow$ Hyperparameter

$w_0 \; x_1 \; x_2 \; \cdots \; x_n \; | \; y$ → $(n+1)$

$w_1 \; w_2 \; \cdots \; w_n$

m rows ↓

Ridge

$$L = \sum_{i=1}^{m} (y_i - \bar{y}_i)^2$$

$$= (Xw - y)^T (Xw - y)$$

$y = \begin{bmatrix} - \\ - \\ - \\ - \end{bmatrix}$ m values

$w = \begin{bmatrix} w_0 \\ w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix}$ $(n \times 1)$

$X = \begin{bmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1n} \\ 1 & x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & & & & \\ 1 & x_{m1} & x_{m2} & \cdots & x_{mn} \end{bmatrix}$

Normal linear regression → Ridge

$L = (Xw - y)^T (Xw - y) + \lambda \|w\|^2$

$\lambda w_1^2 + \lambda w_1^2 + \lambda w_2^2 + \dots + \lambda w_n^2$

$\lambda (w_0^2 + w_1^2 + w_2^2 \dots w_n^2)$

$[w_0\ w_1\ w_2 \dots w_n] \begin{bmatrix} w_0 \\ w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix} = w^T w$

$(a-b)^T = a^T - b^T$

$(ab)^T = b^T a^T$

$\frac{dL}{dw} = ?$

$L = (Xw - y)^T (Xw - y) + \lambda w^T w$

$L = [(Xw)^T - y^T] (Xw - y) + \lambda w^T w$

$= (w^T X^T - y^T)(Xw - y) + \lambda w^T w$

$= w^T X^T X w - w^T X^T y - y^T X w + y^T y + \lambda w^T w$

$L = w^T X^T X w - 2 w^T X^T y + y^T y + \lambda w^T w$

$\frac{dL}{dw} = 2X^T X w - 2X^T y + 0 + 2\lambda w = 0$

$X^T X w + \lambda w = X^T y$

$(X^T X + \lambda I) w = X^T y$

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

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

Soln. for Ridge regression

Vectorized form Loss

$$L = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

vectorized form $L = (Xw - y)^T (Xw - y) + \lambda \|w\|^2$

$$L = (Xw - y)^T (Xw - y) + \lambda w^T w$$

$$X = \begin{bmatrix} 1 & x_{11} & x_{12} & x_{13} & \cdots & x_{1n} \\ 1 & x_{21} & x_{22} & x_{23} & \cdots & x_{2n} \\ \vdots & & & & & \\ 1 & x_{m1} & x_{m2} & x_{m3} & \cdots & x_{mn} \end{bmatrix} \quad \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix} \quad \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_n \end{bmatrix}_{(n+1)}$$

$w_0, w_1, \ldots, w_n$ (parameters)

$$w_0 = w_0 - \eta \frac{\partial L}{\partial w_0}, \quad w_1 - \eta \frac{\partial L}{\partial w_1}, \quad \cdots \quad w_n = w_n - \eta \frac{\partial L}{\partial w_n}$$

~~$\Delta w = w_0$~~ $w_{new} = w_{old} - \eta \left[\frac{\partial L}{\partial w} \right]$ → Gradient

$$\begin{bmatrix} \frac{\partial L}{\partial w_0} \\ \partial L / \partial w_1 \\ \vdots \\ \partial L / \partial w_n \end{bmatrix}$$

L ; w

$$L = \frac{1}{2}(Xw - y)^T (Xw - y) + \frac{\lambda}{2} w^T w$$

$$= \frac{1}{2}(w^T X^T - y^T)(Xw - y) + \frac{\lambda}{2} w^T w$$

$\frac{1}{2}\left[\omega^T X^T X\omega - \omega^T X^T y - y^T X\omega + y^T y\right] + \frac{1}{2}\lambda\omega^T\omega$

$= \frac{1}{2}\left[\omega^T X^T X\omega - 2\omega^T X^T y + y^T y\right] + \frac{1}{2}\lambda\omega^T\omega$

$= \frac{dL}{d\omega} = \frac{1}{2}\left[2X^T X\omega + 2X^T y\right] + \frac{1}{2}2\lambda\omega$

$= X^T X\omega - X^T y + \lambda\omega = \frac{dL}{d\omega}\left(\frac{\Delta L}{\Delta\omega}\right)$

$\omega = [\omega_0\ \omega_1 \cdots \omega_m]$ starting

$[0, 1 \cdots n_m]$

epochs $\omega = \omega - \eta\frac{dL}{d\omega}$

$\omega \rightarrow$ final answer

$\omega = \omega - \eta\frac{dL}{d\omega}$ epochs times

$$\frac{dL}{d\omega} = X^T X\omega - X^T y + \lambda\omega$$

Lasso Regression (L1)

$L = MSE + \lambda\|\omega\|$

$\lambda\left[|\omega_1| + |\omega_2| + \cdots |\omega_n|\right]$

code, implement | sklearn

→ eg point

→ difference Ridge vs Lasso regression

Simple $y = mx + b$

$b = \bar{y} - m\bar{x}$

$\bar{y} = \text{mean}(y)$

$\bar{x} = \text{mean}(x)$

$$m = \frac{\sum_{i=1}^{n} (y_i - \bar{y})(x_i - \bar{x})}{\sum_{i=1}^{n} (x_i - \bar{x})^2}$$

$$m = \frac{\sum_{i=1}^{n} (y_i - \bar{y})(x_i - \bar{x})}{\sum_{i=1}^{n} (x_i - \bar{x})^2 + \lambda}$$

mode m is not differentiable in two ways

$b = \bar{y} - m\bar{x}$ $\quad m = ?$

Loss $L = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 + \lambda |m|$ for mathematical simplicity

$$\frac{dL}{dm} = \sum_{i=1}^{n} (y_i - mx_i - \bar{y} + m\bar{x})^2 + \lambda |m|$$

$m > 0$

$|m| = m$

$$\frac{d}{dm} \sum_{i=1}^{n} (y_i - mx_i - \bar{y} + m\bar{x})^2 + 2\lambda m$$

$$2 \sum (y_i - mx_i - \bar{y} - m\bar{x})(-x_i + \bar{x}) + 2\lambda = 0$$

$$-2 \sum [(y_i - \bar{y}) - m(x_i - \bar{x})](x_i - \bar{x}) + 2\lambda = 0$$

$$- \sum [(y_i - \bar{y})(x_i - \bar{x}) - m(x_i - \bar{x})^2] + 2\lambda = 0$$

$$- \sum (y_i - \bar{y})(x_i - \bar{x}) + m \sum (x_i - \bar{x})^2 + 2\lambda$$

$$m\sum(x_i-\bar{x})^2 = \sum(y_i-\bar{y})(x_i-\bar{x}) + \lambda$$

$$m = \frac{\sum(y_i-\bar{y})(x_i-\bar{x}) + \lambda}{\sum(x_i-\bar{x})^2}$$

Lasso

for $m > 0$

$$m = \frac{\sum(y_i-\bar{y})(x_i-\bar{x}) - \lambda}{\sum(x_i-\bar{x})^2}$$

for $m = 0$

$$m = \frac{\sum(y_i-\bar{y})(x_i-\bar{x})}{\sum(x_i-\bar{x})^2}$$

for $m < 0$

$$m = \frac{\sum(y_i-\bar{y})(x_i-\bar{x}) + \lambda}{\sum(x_i-\bar{x})^2}$$

Sparsity occur why?

$m < 0$ $\quad m = \dfrac{\sum(y_i-\bar{y})(x_i-\bar{x}) + \lambda}{\sum(x_i-\bar{x})^2}$

$\lambda > 0$

$m = \dfrac{-100+\lambda}{50}$

$= \dfrac{-100-150}{50}$

$= -5$

$\lambda = 0, \; m = -2$

$\lambda = 50, \; m = -1$

$\lambda = 100, \; m = 0$

$\lambda = 150, \; m = 1$

1) 0 →

2) 0 → stop

why Ridge regression no sparsity

$$m = \frac{\sum(y_i-\bar{y})(x_i-\bar{x})}{\sum(x_i-\bar{x})^2 + \lambda}$$

← Denominator

$\lambda = 10000000$

$m = 0$, when numerator is zero

In ridge λ is numerator

In lasso λ is denominator terms

Elastic Regression

it's a combination of Ridge and Lasso

Ridge

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

MSE — overfitting reduce

used where all feature are important

↳ 100 column.

Lasso

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

- feature selection
- if any feature is not important as compared to other particular feature eliminate that particular feature

If you have large data set and you are unable to decide which one is applied in that case go to elastic net regression

$$L = \Sigma (y_i - \hat{y}_i)^2 + a\|w\|^2 + b\|w\|$$

$$\lambda = a + b$$

$$= \frac{a}{a+b}$$

λ and l1_ratio
two hyperparameters

$\lambda = 1$, l1_ratio $= 0.5$

$a = 0.5$ $b = 0.5$

↑ l1_ratio $= 0.9$
90% ridge regression
10% Lasso regression

L1 = ratio

$$L1 = \frac{a}{\lambda}$$

$$l1 \times \lambda = a \qquad b = \lambda - a$$

Input wt. — Multicollinearity

x_1 | b_1

Logistic Regression

Introduction :- easy method to learn. Requirement ⇒ linearly separable or nearly linearly separable

2 perspective
- geometric
- probability

- If we apply logistic regression on non linear data then we didn't get good result. (y = mx + c linear eqn)
- Perceptron trick

$Ax + By + C = 0$

$Ax_1 + Bx_2 + C = 0$

$Ax_1 + Bx_2 + Cx_3 + d = 0$ ↑ ing dimensions

A, B, C to find these values

Demerits for presence

start random value of A, B, C

- random student, Ask from different random points / transformation ↑ ~~the~~ ~~the~~ ~~on~~

loop → randomly pick point → shifting the line from response of randomly pick point
- 1000 epochs
- till convergence

How to identify true -ve region

How to label a region

Transformation

C ⟹ Line move parallelly upward and downward

A → from point y more

B → from x more.

not good ✗

$2x + 3y + 5 \geq 0$ +ve

$2x + 3y + 5 < 0$ −ve or below

$2x + 3y + 5 = 0$

(4,5) → 4, 5, 1

2 3 5

when line moves to region then subtract

− 4 5 1

$2x - 2y + 4 = 0$

(1, 3) → (1, 3, 1)

2 3 5

+ 1 3 1

$3x + 6y + 6 = 0$

when line moves to −ve region then add

learning rate → 0.01

coef = coef − η coordinates

x_0	x_1 (cgpa)	x_2 (Iq)	y (placed)
1	7.5	81	1
1	8.7	109	1
1	7.0	81	0

$Ax + By + C = 0$

$w_0 + w_1 x_1 + w_2 x_2 = 0$

$w_0 = 0$

$w_1 = A$

$w_2 = B$

for simplicity equation $w_0 x_0 + w_1 x_1 + w_2 x_2 = 0$

$\sum_{i=0}^{2} w_i x_i = 0$

$w_0 x_0 + w_1 \times 7.5 + w_2 \times 81$ ≥ 0 placed

< 0 not placed

$[w_0, w_1, w_2] \begin{bmatrix} x_0 \\ x_1 \\ x_2 \end{bmatrix}$

$w_0 x_0 + w_1 x_1 + w_2 x_2 \to 1$
$\hookrightarrow 0$

decide epoch → 1000
$\eta = 0.01$
for i in range epochs:
random only select a pt. or student.
— if

$w_{new} = w_{old} - \eta [x_0 x_1 x_2]$
$[x_i]$

If $x_i \in P$ and $\sum_{i=1}^{2} w_i x_i < 0$

P = positive region

$w_{new} = w_{old} + \eta x_i$

Simplified the Algo

for i in 1000 random only select student
$w_n = w_0 + \eta (y_i - \hat{y}_i) x_i$

y_i	$\hat{y}_i$	$y_i - \hat{y}_i$	
1	1	0	$w_n = w_{old}$
0	0	0	$w_n = w_{old}$
1	0	1	$w_n = w_0 + \eta x_i$
0	1	−1	$w_n = w_{old} - \eta x_i$

change the previous approach

— misclassified — line pull
correctly classified — line push

$$w_n = w_0 + (y_i - \hat{y}_i) x_i$$

$y = 0, 1$

$-\infty < z < \infty$
$0 < y < 1$ sigmoid function $\frac{1}{1 + e^{(-z)}} = \sigma(z)$

Sigmoid function give probability function

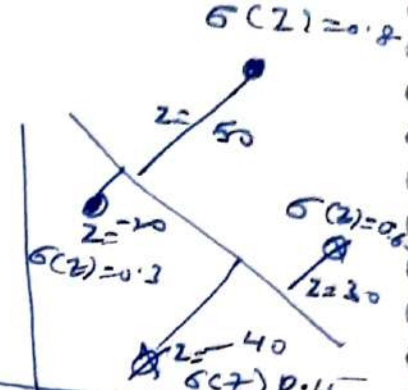
Intersection of P and line will depend on the distance from line to point

$\sigma = 0.5$ $\rho = 0.5$

$$w_n = w_0 + \eta(y_i - \hat{y}_i)x_i$$

$$y_i = \sigma(z)$$

where $z = \sum w_i x_i$



$w_n = w_{old} + \eta * 0.2 * x_i$

$w_n = w_0 - \eta * 0.65 * x_i$

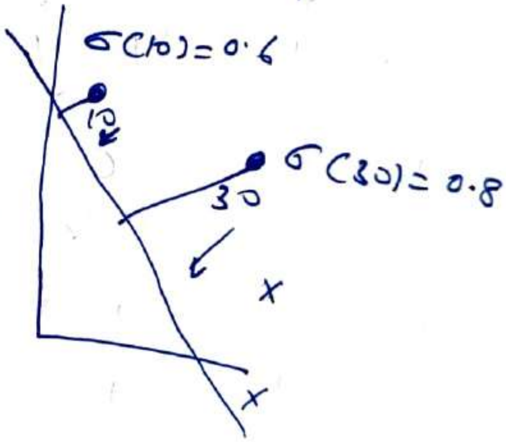
$w_n = w_0 + \eta * 0.7 * x_i$

$w_n = w_0 - \eta * 0.15 * x_i$

y_i	$\hat{y}_i$	$y_i - \hat{y}_i$
1	0.8	0.2
0	0.65	-0.65
1	0.3	0.7
0	0.15	-0.15

Statistic shortest distance parallel straight line perpendicular distance

y_i	$\hat{y}_i$	$y_i - \hat{y}_i$
1	0.6	0.4
	0.8	0.2
1		
0		
0		



$w_n = w_0 + \eta * 0.4 * x_i = x_1$

$w_n = w_0 + \eta * 0.2 * x_i = x_2$ $x_1 > x_2$

Maximum likelihood (based on probability)

$$\hat{y} = \sigma(z)$$

$\to z = \sum w_i x_i$

methodology is not true because of 1000 epochs every time give new result

loss function error predictions

if in real world dataset no. 1's more than predicted is very small

$\log(ab) = \log a + \log b + \log(c \cdot d) + \ldots$

log max: $\log(0.7) + \log(0.4) + \log(0.8)$

0-1 = -ve

convert all logs into -ve $\log(0.7) \quad \log(0.4)$

$\log(\text{max}) = -\log(0.7) - \log(0.4) - \log(0.3) - \log(0.8)$

cross entropy minimize

$-y_i \log(\hat{y}_i) - (1-y_i)\log(1-\hat{y}_i)$

$-y_i \log(\hat{y}_i) = -\log(\hat{y}_1)$

$= -\log(0.7)$

$-y_2 \log(\hat{y}_2) - (1-y_2)\log(1-\hat{y}_2)$ $\quad y = 0$

$= -\log(1-\hat{y}_2) = -\log 0.4$

$y_3 = 1$

$-y_3 \log(\hat{y}_3) = \log(\hat{y}_3) = -\log(0.4)$

$y_4 = 0, \quad (1-y_4)\log(1-\hat{y}_4)$

$= -\log(1-\hat{y}_4)$

$= -\log(0.8)$

$$L = \sum_{i=1}^{4} -y_i \log(\hat{y}_i) - (1-y_i)\log(1-\hat{y}_i)$$

$$L = -\frac{1}{n}\sum_{i=1}^{n} y_i \log(\hat{y}_i) + (1-y_i)\log(1-\hat{y}_i)$$

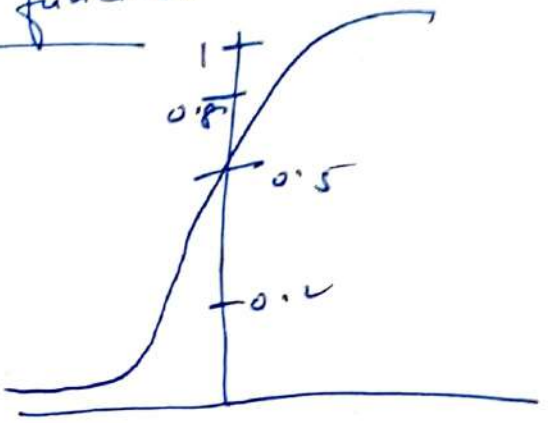
→ log loss error

multi
output

closed form → Binary cross entropy function.

↓ gradient descent.

Derivative of sigmoid function



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

differentiation reciprocal rule

$$\frac{d}{dx}\left(\frac{1}{x}\right) = \frac{d}{dx}(x)^{-1}$$

$$= -x^{-2} = \frac{-1}{x^2}$$

$$\frac{d}{dx}\left[\frac{1}{1+e^{-x}}\right] = \frac{d}{dx}\left[1+e^{-x}\right]^{-1}$$

$$= -\frac{1}{(1+e^{-x})^2}\frac{d}{dx}(1+e^{-x})$$

chain rule

$e^{-x} = e^{x}$

$$= \frac{-1}{(1+e^{-x})^2}\frac{d}{dx}(e^{-x}) = \frac{-e^{-x}}{(1+e^{-x})^2}\frac{d}{dx}(-x)$$

$$= \frac{e^{-x}}{(1+e^{-x})^2}$$

$$\frac{1 \cdot e^{-x}}{(1+e^{-x})(1+e^{-x})} = \frac{1}{1+e^{-x}} \cdot \frac{e^{-x}}{1+e^{-x}} = \sigma(x)\left[\frac{e^{-x}}{1+e^{-x}}\right]$$

Sunnat@2022

$$= \sigma(x)\left[\frac{1+e^{-x}-1}{1+e^{-x}}\right] = \sigma(x)\left[\frac{1+e^{-x}}{1+e^{-x}} - \frac{1}{1+e^{-x}}\right]$$

$$\sigma(x)\left[1-\sigma(x)\right] \Rightarrow \sigma'(x) = \sigma(x)\left[1-\sigma(x)\right]$$

L-6.7.4 Logestic Regression - Gradient descent

Rows = m Column = n

1	2	3		n	y
x_{11}	x_{12}	x_{13}		x_{1n}	y_1
x_{21}	x_{22}	x_{23}		x_{2n}	y_2
⋮					
x_{m1}	x_{m2}	x_{m3}		x_{mn}	y_m

$\hat{y}$ $\{w_1\ w_2\ w_3 \cdots w_n\}$ w_0 $(n+1)$

1st row

$\sigma(w_1 x_1 + w_2 x_{12} + w_3 x_{13} + \cdots + w_n x_{1n} + w_0) = \hat{y}_1$ ...

2nd row

$\sigma(w_1 x_{21} + w_2 x_{22} + w_3 x_{23} + \cdots + w_n x_{2n} + w_0) = \hat{y}_2$

$$\hat{y} = \begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \vdots \\ \hat{y}_m \end{bmatrix} = \begin{bmatrix} \sigma(w_0 + w_1 x_{11} + w_2 x_{12} + \cdots + w_n x_{1n}) \\ \sigma(w_0 + w_1 x_{21} + w_2 x_{22} + \cdots + w_n x_{2n}) \\ \vdots \\ \sigma(w_0 + w_1 x_{m1} + w_2 x_{m2} + \cdots + w_n x_{mn}) \end{bmatrix}$$

$$\hat{y} = \sigma\left(\begin{bmatrix} 1 & w_0 + w_1 x_{11} + w_2 x_{12} + \cdots + w_n x_{1n} \\ 1 & w_0 + w_1 x_{21} + w_2 x_{22} + \cdots + w_n x_{2n} \\ \vdots & \\ 1 & w_0 + w_1 x_{m1} + w_2 x_{m2} + \cdots + w_n x_{mn} \end{bmatrix} \right)$$

w_0

dot product

$$\hat{y} = \sigma\left(\begin{bmatrix} x_{11} & \cdots & x_{1n} \\ x_{21} & \cdots & x_{2n} \\ x_{m1} & \cdots & x_{mn} \end{bmatrix} \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_n \end{bmatrix} \right)$$

$$\hat{y} = \sigma(xw)$$

$$L = -\frac{1}{m}\sum_{i=1}^{m} y_i \log(\hat{y}_i) + (1-\hat{y}_i)\log(1-\hat{y}_i)$$

$$L = -\frac{1}{m}\left[\sum_{i=1}^{m} y_i \log(\hat{y}_i) + \sum_{i=1}^{m}(1-y_i)\log(1-\hat{y}_i)\right]$$

$$\sum_{i=1}^{m} y_i \log(\hat{y}_i) = y_1 \log \hat{y}_1 + y_2 \log \hat{y}_2 + y_3 \log \hat{y}_3 + \cdots + y_m \log \hat{y}_m$$

$$\begin{bmatrix} y_1 & y_2 & y_3 & \cdots & y_m \end{bmatrix} \begin{bmatrix} \log \hat{y}_1 \\ \log \hat{y}_2 \\ \log \hat{y}_m \end{bmatrix}$$

$$\begin{bmatrix} y_1 & y_2 & y_3 & \cdots & y_m \end{bmatrix} \log\left(\begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \hat{y}_m \end{bmatrix}\right) \quad \Longrightarrow$$

$$y \log \hat{y} = y \log(\sigma(xw))$$

$$L = -\frac{1}{m}\left[y \log \hat{y} + (1-y)\log(1-\hat{y})\right]$$

where $\hat{y} = \sigma(xw)$ ↑ gradient descent [w] find

Loss function in matrix form

$$L = \frac{1}{m}\left[y \log(\sigma(wx)) + (1-y)\log(1-\sigma(wx))\right]$$

↑ minimum

gradient descent

for i in epoch:

$w = w - \eta \frac{\Delta L}{\Delta w}$

$w = [\quad]$

$w_0 \rightarrow w_n$

$\left[\frac{\Delta L}{\Delta w}\right]$ $\qquad \frac{\Delta L}{\Delta w} = \left[\frac{\partial L}{\partial w_0}, \frac{\partial L}{\partial w_1}, \frac{\partial L}{\partial w_2} \cdots \frac{\partial L}{\partial w_n}\right]$ ← (n+1)

$L = -\frac{1}{m}\left[y \log \hat{y} + (1-y)\log(1-\hat{y})\right]$

$\frac{dL}{dw} = \frac{d}{dw} y \log \hat{y} \Rightarrow y \frac{d}{dw} \log \hat{y} \Rightarrow \frac{y}{\hat{y}} \frac{d}{dw}(\hat{y})$

$\Rightarrow \frac{y}{\hat{y}} \frac{d}{dw} \sigma(wx) \Rightarrow \frac{y}{\hat{y}} \sigma(wx)[1-\sigma(wx)] \frac{d}{dw}(wx)$ ($\hat{y} = \sigma(wx)$)

$= \frac{y}{x} x(1-\hat{y})x = \boxed{y(1-\hat{y})x}$

$\frac{d}{dw}(1-y)\log(1-\hat{y}) \Rightarrow (1-y)\frac{d}{dw}\log(1-\hat{y}) \Rightarrow$

$\Rightarrow \frac{(1-y)}{(1-\hat{y})} \frac{d}{dw}(1-\hat{y})$ $\qquad \{\hat{y} = \sigma(wx)$

$= -\frac{(1-y)}{(1-\hat{y})} \frac{d}{dw} \sigma(wx) \Rightarrow -\frac{(1-y)}{(1-\hat{y})}[\sigma(wx)][1-\sigma(wx)] \frac{d}{dw}(wx)$

$\Rightarrow -\frac{(1-y)}{(1-\hat{y})} \hat{y}(1-\hat{y})x = \boxed{-\hat{y}(1-y)x}$

$$\frac{dL}{dw} = -\frac{1}{m}\left[y(1-\hat{y})x - \hat{y}(1-y)x\right]$$

$$= -\frac{1}{m}\left[y(1-\hat{y}) - \hat{y}(1-y)\right]x$$

$$= -\frac{1}{m}\left[y - y\hat{y} - \hat{y} + y\hat{y}\right]x$$

$$\boxed{\frac{\Delta L}{\Delta w} = -\frac{1}{m}(y-\hat{y})x}$$

gradient descent update →

$$\boxed{w = w + \eta\frac{1}{m}(y-\hat{y})x}$$

$w = \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_n \end{bmatrix}_{(n+1,1)}$

$x = \begin{bmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1n} \\ 1 & x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & & & & \\ 1 & x_{m1} & x_{m2} & \cdots & x_{mn} \end{bmatrix}_{(m,n+1)}$

$y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix}_{(m,1)}$

$\hat{y} = \begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \vdots \\ \hat{y}_m \end{bmatrix}_{(m,1)}$

$w = w + \frac{\eta}{m}(y-\hat{y})x$

$(n+1,1) \quad (m,1) \quad (m,n+1)$

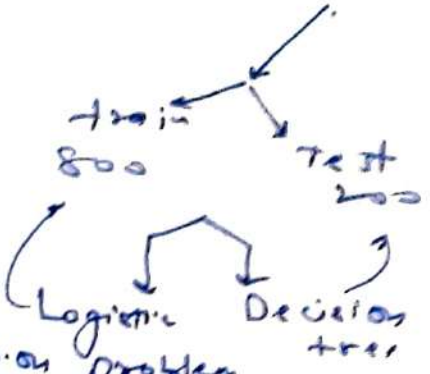
$(1,m)$ ← transpose

Accuracy and confusion matrix

Accuracy

cgpa | iq | placement

accuracy = no. of correct predictions / total no. of predictions = 1000



$\frac{8}{10} = 0.8 = 80\%$ accurate

Accuracy of multi classification problem

How much accuracy is good:-

it depend the problem we are solving

eg in medical field. 100% accuracy is needed
even 99% accuracy is suppose to be bad

- self driving car 100% accuracy is required

whereas Swiggy for food recommendation 80% is good

90% accurate → 10% wrong.

Accuracy score not tell us about type of mistake (nature of mistake)

Confusion matrix : (predict nature of mistake)

Actual \ Predicted	1	0
1	TP 26	Type II error FN 6
0	Type I error FP 0	TN 29

accuracy = # of correct predictions / # of predictions

$$acc = \frac{TP + TN}{TP + TN + FP + FN}$$

Type 1 Error / false positive
Type 2 Error false -ve

confusion matrix for multiclass classification problem

Predicted / Actual	0	1	2
0	7	0	1
1	2	21	6
2	1	1	17

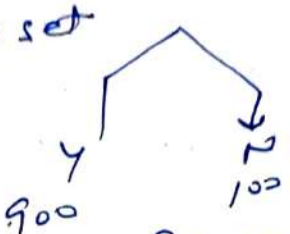
3 × 3

Diagonal elements are true predictions

$\frac{\text{true pred.}}{\text{total}}$ = accuracy

when accuracy is misleading

1. imbalanced data set



when cost is not appropriate

Precision, Recall and f1 score

Precision

$FP_a > FP_b$ | $FN_a < FN_b$

what proportion of predicted positive is truly positive

called precision

$$\text{Precision} = \frac{TP}{(TP + FP)}$$

Recall :- what proportion of actual +ve is correctly classified

$$\text{Recall} = \frac{TP}{TP + FN}$$

$$\text{Recall A} = \frac{1000}{1200}$$

f1 score :- Its is combination of precision and recall.

Harmonic mean toward lower side

$$f1\ score = \frac{2PR}{P+R}$$

- macro precision calculate: $\frac{0.86 + 0.66 + 0.58}{3} = 0.70$
- weighted precision: $\left(\frac{40}{108} 0.86\right) + \left(\frac{34}{108} 0.64\right) + \left(\frac{34}{108} 0.58\right)$

$= 0.71$

If your classes balance then used macro
If your classes imbalance then used weighted

$R_D = \frac{25}{40} = 0.62$, $R_C = \frac{30}{34} = 0.88$

$R_R = \frac{34}{20} = 0.58$

- macro recall
- weighted recall

$$f1_D = \frac{2 P_D R_D}{P_D + R_D} = \frac{2 \times 0.86 \times 0.62}{0.86 + 0.66}$$

Softmax Regression (Multinomial Regression)

cgpa	iq	plcd
7.1	71	0
8.5	85	1
9.5	95	2

yes 1
no 0
optout 2

{6.5, 6.5} → yes, no, optout

softmax → general → for 2 classes

softmax function

$$\sigma(z)_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}$$ K = no of classes

yes = 1
no = 0
apna = 1

$$\sigma(z)_1 = \frac{e^{z_1}}{e^{z_1} + e^{z_2} + e^{z_3}} \qquad \sigma(z)_2 = \frac{e^{z_2}}{e^{z_1} + e^{z_2} + e^{z_3}}$$

$$\sigma(z_3) = \frac{e^{z_3}}{e^{z_1} + e^{z_2} + e^{z_3}}$$ value is b/w 0 to 1 and adding all three equal to 1

Training intuition

cgpa	iq	placement?
–	–	0
–	–	1
–	–	2
–		

→ transform your data one hot encode

cgpa	iq	z=0	z=1	z=2
		1	0	0
		0	1	0
		0	0	1

D

D1 D2 D3

D1: cgpa | iq | = 0 → M1

D2: cgpa | iq | = 1 → M2

D3: cgpa | iq | = 2 → M3

3 coeff

w_1, w_2, w_0

m_1 yes

$w_1^{(0)}, w_2^{(0)}, w_0^{(0)}$

m_2 no

$w_1^{(1)}, w_2^{(1)}, w_0^{(1)}$

m_3 op+out

$w_1^{(2)}, w_2^{(2)}, w_0^{(2)}$

$$z_1 = 7 \times w_1^{(1)} + 7 \times w_2^{(2)} + w_0$$

$$z_2 = 7 \times w_1^{(2)} + 7 \times w_2^{(2)} + w_0$$

$$z_3 = 7 \times w_1^{(3)} + 7 \times w_2^{(3)} + w_0^{(3)}$$

$$\sigma(y) = \frac{e^{z_1}}{e^{z_1} + e^{z_2} + e^{z_3}}$$

(0.40)

$$G(1) = \frac{e^{z_2}}{e^{z_1} + e^{z_2} + e^{z_3}}$$

(0.35)

$$c(0) = \frac{e^{z_3}}{e^{z_1} + e^{z_2} + e^{z_3}}$$

(0.25)

Loss function

$$L = \frac{1}{m} \sum_{i=1}^{m} y_i \log(\hat{y}_i) + (1-\hat{y}_i) \log(1-\hat{y}_i)$$

$$L = -\frac{1}{m} \sum_{i=1}^{m} \sum_{k=1}^{K} y_k^{(i)} \log(\hat{y}_k^{(i)})$$

x_1	x_2	y	$y_{k=1}$	$y_{k=2}$	$y_{k=2}$
x_{11}	x_{12}	0	1	0	0
x_{21}	x_{22}	2	0	0	0
x_{31}	x_{32}	3			

$$y_0^{(1)} \log(\hat{y}_1^{(1)}) + y_1^{(1)} \log(\hat{y}_2^{(1)}) + y_2^{(1)} \log(\hat{y}_3^{(1)})$$
$$+ y_1^{(2)} \log(\hat{y}_1^{(2)}) + y_2^{(2)} \log(\hat{y}_2^{(2)}) + y_3^{(2)} \log(\hat{y}_3^{(2)})$$
$$+ y_1^{(3)} \log(\hat{y}_1^{(3)}) + y_2^{(3)} \log(\hat{y}_2^{(3)}) + y_3^{(3)} \log(\hat{y}_3^{(3)})$$

$$L = y_1^{(1)} \log(\hat{y}_1^{(1)}) + y_2^{(1)} \log(\hat{y}_2^{(1)}) + y_3^{(1)} \log(\hat{y}_3^{(1)})$$

$\hat{y}_1^{(1)}, \hat{y}_2^{(1)}, \hat{y}_3^{(1)} = ?$

$\hat{y}_1^{(1)} = \sigma(w_1^{(1)} x_{11} + w_2^{(1)} x_{12} + w_0^{(1)})$ → softmax

$\hat{y}_2^{(2)} = \sigma(w_1^{(2)} x_{11} + w_2^{(2)} x_{12} + w_0^{(2)})$

$\hat{y}_3^{(3)} = \sigma(w_1^{(3)} x_{11} + w_2^{(3)} x_{12} + w_0^{(3)})$

$$\begin{bmatrix} w_1^{(1)} & w_2^{(1)} & w_0^{(1)} \\ w_1^{(2)} & w_2^{(2)} & w_0^{(2)} \\ w_1^{(3)} & w_2^{(3)} & w_0^{(3)} \end{bmatrix}$$

$\frac{\partial L}{\partial w_1^{(1)}}, \frac{\partial L}{\partial w_2^{(1)}}, \frac{\partial L}{\partial w_0^{(1)}}$ → gradient

loop → 100 epochs

$w_1^{(1)} = w_1^{(1)} - \eta \frac{\partial L}{\partial w_1^{(1)}}$

learning rate = 0.1

Polynomial Logistic Regression

- multinear data
- Logistic Regression

Polynomial, degree

- useful in non-linear data
- convert non linear to linear by applying poly — non-linear features.

x_0, x_1

x_0^0 x_0^1 x_0^2 x_1^0 x_1^1 x_1^2

— — — — — —

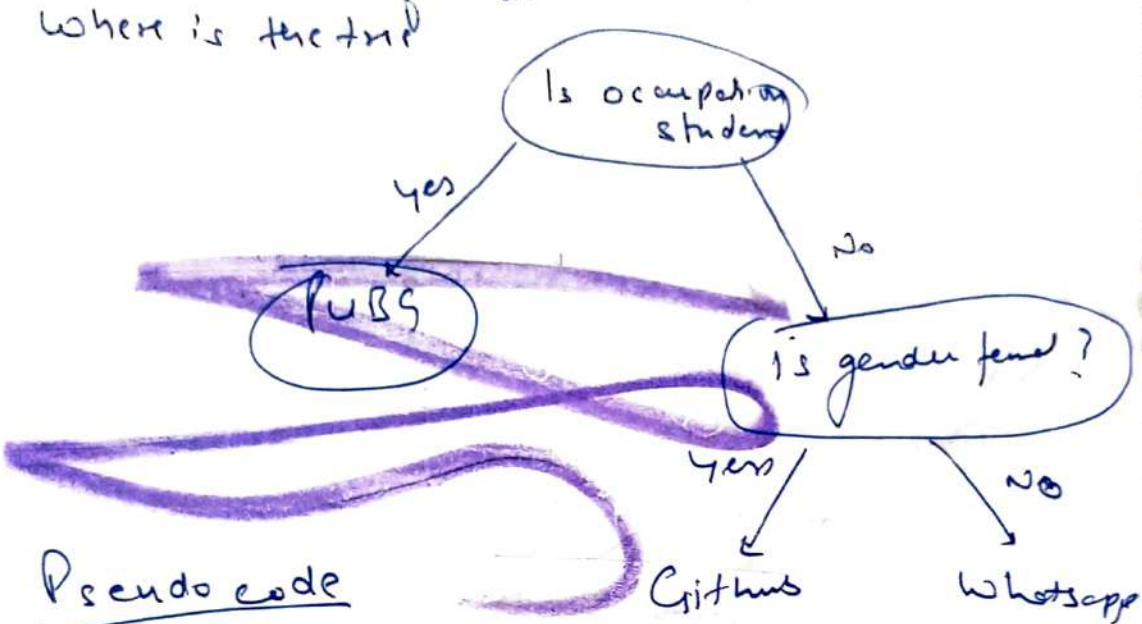
w_0 w_1 w_2 w_3 w_4 w_5 w_6

Logistic hyperparameters

Decision tree Geometric Intuition

nested if else condition

Where is the true



Pseudo code

- Begin with your training dataset, which should have some features variables and classification or regression output
- Determine the best feature in the dataset to split the data on; more on how we define "best feature" later
- Split the data into subset that contains the correct values for this best feature. this splitting basically defines a node on the tree i.e each node is splitting point based on certain feature from our data
- Recursively generate a new tree node by using the subset of data created from step 3

Root node — splitting — Branch (subtree) — Decision tree — Leaf node

Conclusion: Programatically speaking, Decision trees are nothing but a grand structure of nested if-else condition.

Mathematically speaking, Decision trees use hyperplanes which run parallel to any one of the axes to cut your coordinate system into hyper cuboids.

Root node splitting Branch / subtree.

Decision node, Leaf node

Some unanswered question

How to decide which column should be considered as root node?

How to select subsequent decision nodes?

How to decide splitting criteria in case of numerical columns?

CART – Classification and Regression Trees

The logic of DT can be also be applied to regression problems, hence the name CART

Decision tree Entropy

What is entropy the measure of disorder or measure of purity / impurity

How to calculate entropy?

$$E(s) = \sum_{i=1} - P_i \log_2 P_i$$

$$E(D) = -P_{yes} \log_2(P_{yes}) - P_{no} \log_2(P_{no})$$

calculating entropy for 3 class

$$H(X) = -P_y \log_2(P_y) - P_n \log_2(P_n) - P_m \log_2(P_m)$$

Observations

- More the uncertainty more is entropy
- For a 2 class problem the min entropy is 0 and the max is 1
- For more than 2 classes, the min entropy is 0 but the max can be greater than 1
- Both $\log_2$ or $\log_e$ can be used to calculate entropy

Entropy Vs probability

Entropy for continuous probability

Information Gain is a metric used to train Decision trees. Specifically, this metric measures the quality of a split.

The information gain is based on the decision tree's entropy after a dataset is split on an attribute. Constructing a decision tree is all about finding attribute that returns the highest information gain.

Information gain = $\in$(Parent) − {weighted Average} * $\in$(children)

Gini Impurity

$$E = -P_y \log_2(P_y) - P_N \log_2(P_N)$$

$$G_I = 1 - (P_y^2 + P_N^2)$$

P_y — wt of chicks

computation is fast

Handly Numerical data

1. Sort the data on the basis of numerical column.
2. Split the entire data on the basis of every value of user_rating

(5) max [IG1, IG2, IG3 — , IGn]

(6) Do this recursively until you get all the leaf nodes.

Overfitting — Perform well on training, Perform badly on test

max_depth = None :- Perform till leaf not is not achieved

Underfitting

max_depth = None :- only one split

By tuning the hyperparameters to avoid underfitting and overfitting

Hyperparameter | overfitting, underfitting

PCA Complete Machine learning

Principle component analysis

__/__/____

feature extraction → COD curse of Dimensionality

eg photographer taken photo at different angles.

3D movement in 2D image

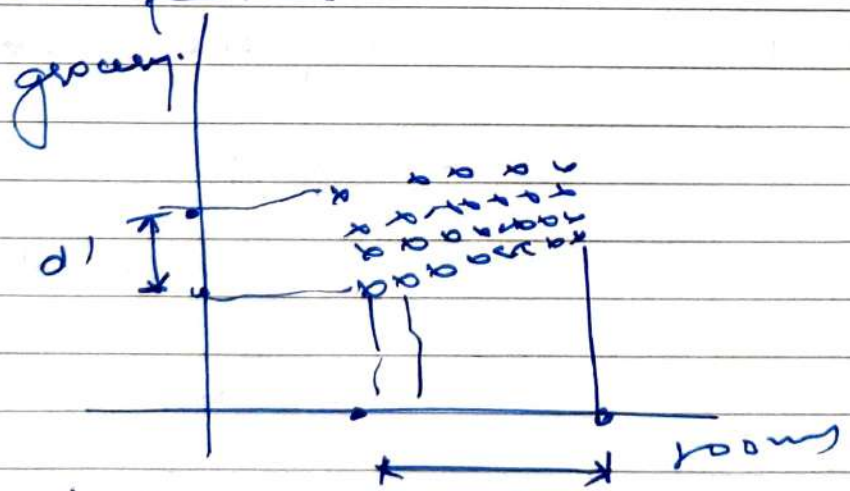
Benifit of PCA:

1. Higher dimensions to lower dimensions
2. Faster execution of algo
3. Visualization

Geometric intuition :-

No. of rooms	No. of grocery shop	Price
3	2	60
4	0	130
5	6	170
2	10	90

feature selection: Reduce the feature only selected important feature



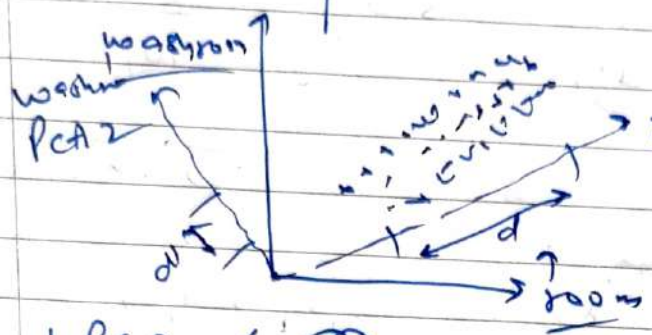
- pick that feature on which variance is high.
- select the column where spread is more

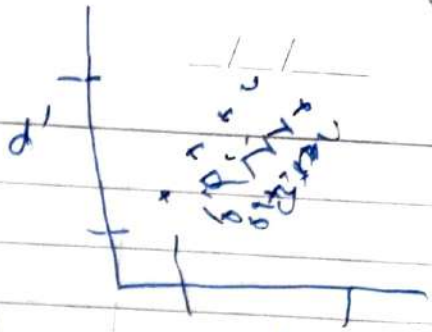
2D data

No. of rooms	No. of washrooms	Price

1D data

Size of flat	Price



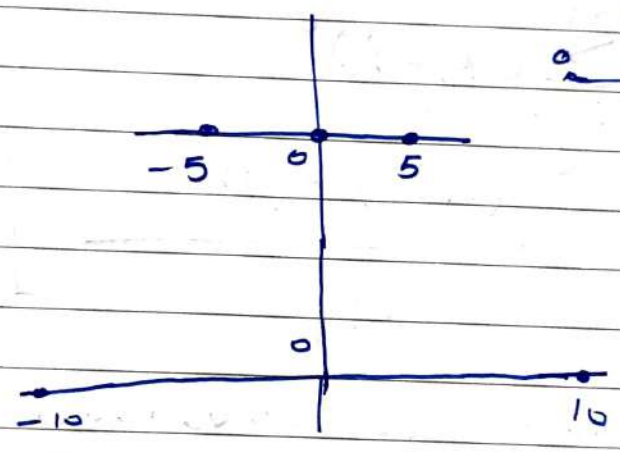


$d \approx d'$

feature extraction

no. of PCA $\leq$ (n)

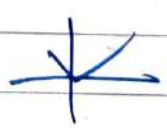
Variance spread → imp?



$$\text{Variance} = \frac{\sum_{i=1}^{n} (x_i - \bar{x})^2}{n} = \frac{25 + 0 + 25}{3} = \frac{50}{3}$$

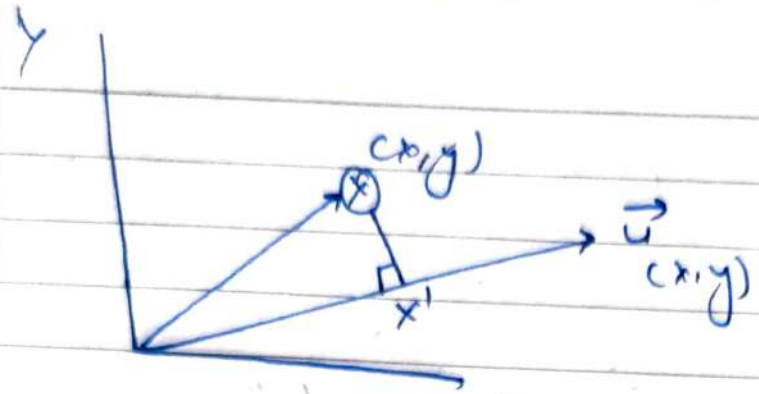
$$\frac{100 + 0 + 100}{3} = \frac{200}{3}$$

Spread ∝ to variance



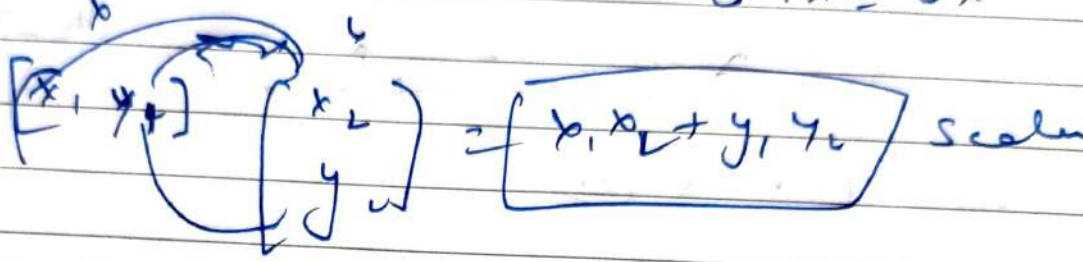
Why variance is important?

Problem formulation 2D → 1D



$\frac{\vec{u} \cdot \vec{x}}{|u| = 1}$

$\vec{u} \cdot \vec{x} = u^T x$



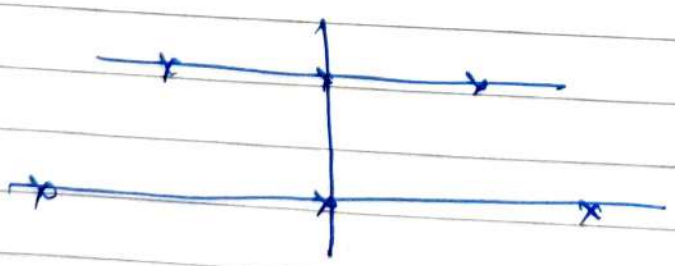
Select unit vector where variance is maximum

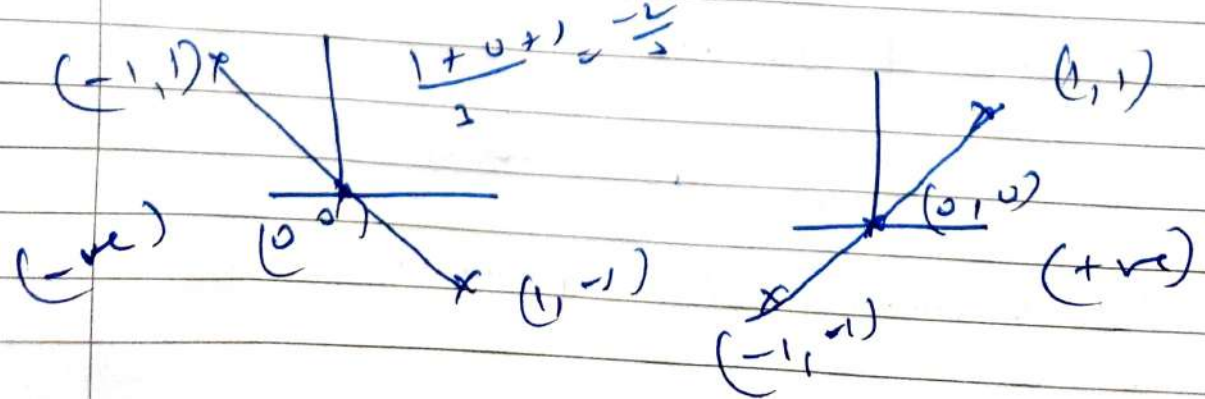
$$\frac{1}{n} \sum_{i=1}^{n} (x_i - \bar{x})^2$$

$$\left[\frac{\sum_{i=1}^{n} (u^T x_i - u^T \bar{x})^2}{n} \right] = \text{variance}$$

→ optimization

Covariance and covariance matrix





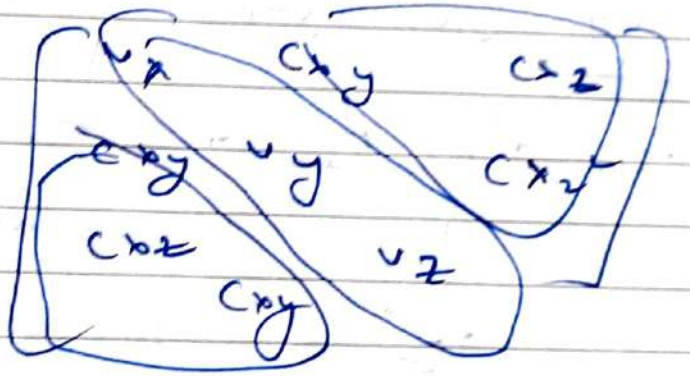
x_1 | x_2 covmatrix

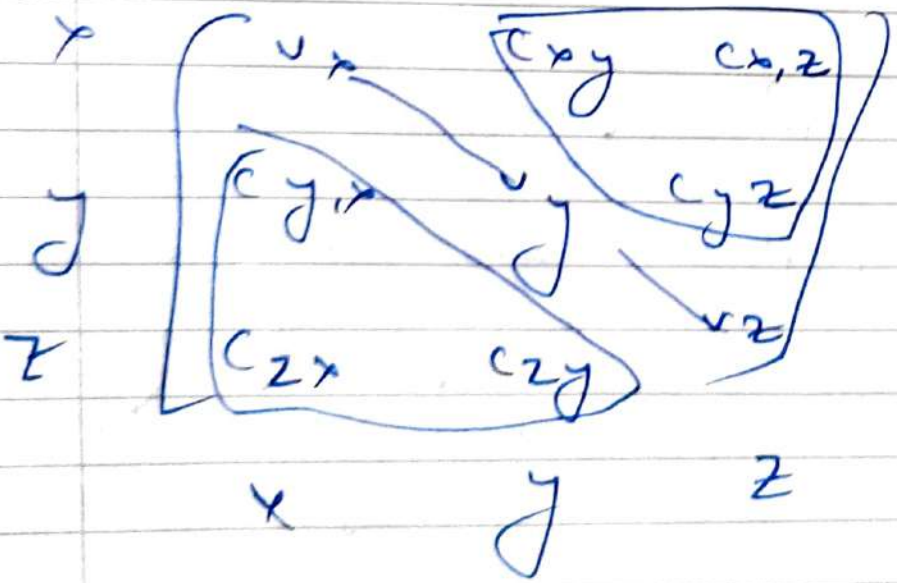
[2 × 2] x_1 [cov(x_1,x_1) cov(
x_2 [cov(x_1,x_2) cov(x_2,x_2)]

cov(x_1, x_1)
↓
cov(x_1)

[Var x_1 cov(x_2,x_1)
cov(x_1,x_2) Var x_2] → square symmetric

cov(a, b) = cov(b, a)





eigen decomposition of covariance matrix

eigen vectors and eigen values.

$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ Eigenvector are special vector whose direction not change but magnitude change

Eigenvalues How much eigenvector shrink or transform

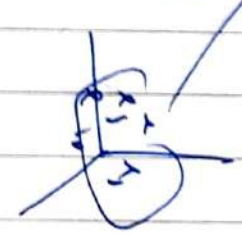
$$A\vec{V} = \lambda\vec{V}$$

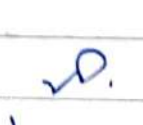
eigenvalue

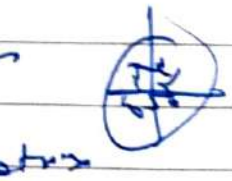
5. Step by step ish

f_1 f_2 f_3

3D







1. mean center
2. find cov. matrix

$$\begin{array}{c} \\ f_1 \\ f_2 \\ f_3 \end{array} \begin{bmatrix} v_{f_1} & c_{f_1 f_2} & c_{f_1 f_3} \\ c_{(f_1 f_2)} & v_{f_2} & c_{(f_2 f_3)} \\ c_{(f_3 f_4)} & & v_{f_3} \end{bmatrix} \begin{array}{l} \\ f_1 \\ f_2 \\ f_3 \end{array}$$

cov matrix

cov matrix

3D → 3 vector

λ_1 λ_2 λ_3

PC1 PC2 PC3

10

20

How to transform point:-

~~f1 | f2 | f3 | target~~

x	y	target
(1000, 3)	(1, 3)	(3, 1)

PC1	target
(1000, 3)	(3, 1)
(1000, 1)	

find optimum number of PCA

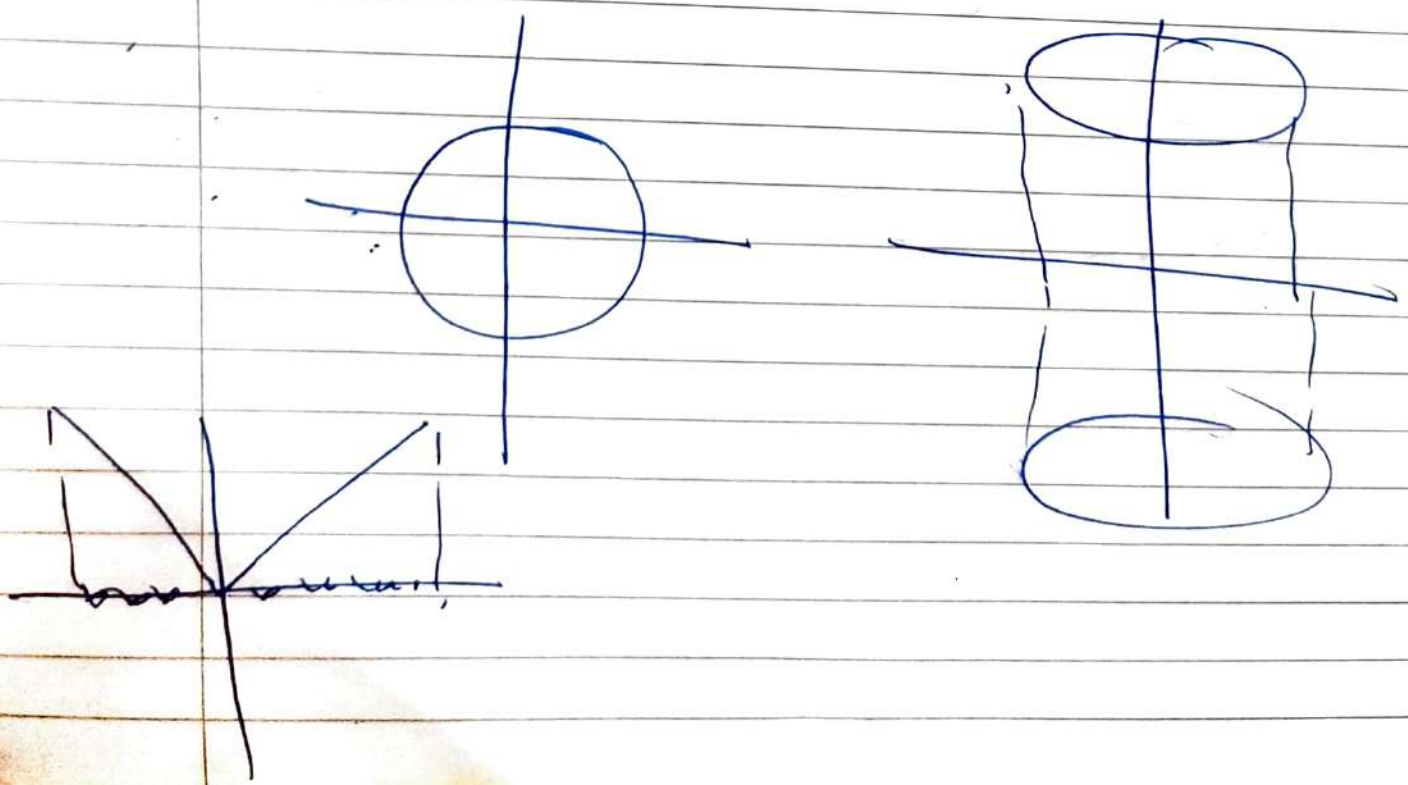
$$Cv = \lambda$$

$$\text{Eig} = \lambda_1, \lambda_2, \lambda_3 \dots \lambda_{\text{Eig}}$$

$$\frac{\lambda_1}{\lambda_1 + \lambda_2 + \lambda_3 + \dots + \lambda_{\text{Eig}}} \times 100 \longrightarrow \text{percentage}$$

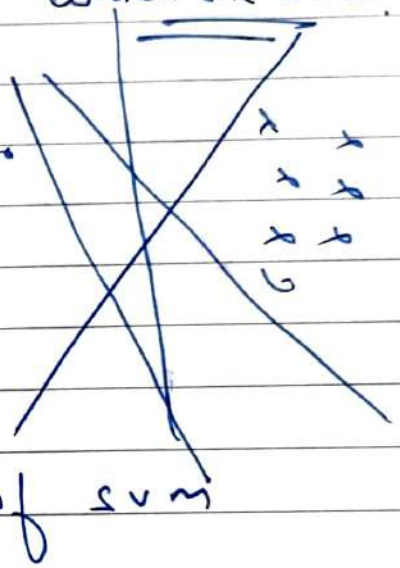
90%.

Q When PCA do not work?



Support vector machine — Ahlad Kumar

- concept of Margin
- which one to select
- middle of these two datasets
- Margin is maximize from both sides of the dataset
- maximum distance from closest point



Theory of svm

Max-Margin ← objective of svm

Question to be asked

1. How to measure the distance from a pt. to a plan
2. How to formalize a max-margin problem
3. How to solve max-margin problem

— Max margin

$$g_\theta = x'\theta + \theta_0$$

theorem Let $g_\theta(x) = \theta^T x + \theta_0$ be a linear discriminant function

Let $H = \{x \mid g(x) = 0\}$ be separating hyperplane

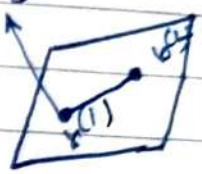
then

i. $\theta / \|\theta\|_2$ is the normal vector of H i.e

for any $x^{(1)}, x^{(2)} \in H$ we have $\theta^T (x^{(1)} - x^{(2)}) = 0$

(ii) for any $x^{(0)}$, the distance b/w $x^{(0)}$ and H is

$$d(x^{(0)}, H) = \frac{g_\theta(x^{(0)})}{\|\theta\|_2}$$



Proof:- (i) We consider two point $x^{(1)} \in H$, $x^{(2)} \in H$ since both the point are on the hyperplane, we have

$g_\theta(x^{(1)}) = 0$ & $g_\theta(x^{(2)}) = 0$ and hence

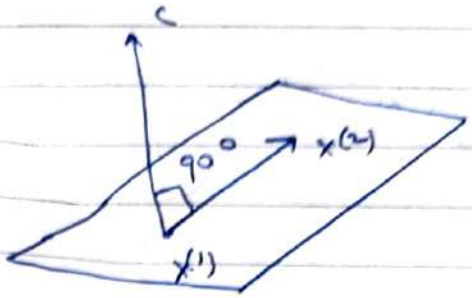
~~$\theta^T (x^{(1)}) = 0$~~

$$\left.\begin{aligned} &\theta^T x^{(1)} + \theta_0 = 0, \quad \theta^T x^{(2)} + \theta_0 = 0 \\ \rightarrow\ &\theta^T x^{(1)} + \theta_0 = \theta^T x^{(2)} + \theta_0 \\ \Rightarrow\ &\theta^T (x^{(1)} - x^{(2)}) = 0 \end{aligned}\right\} (*)$$

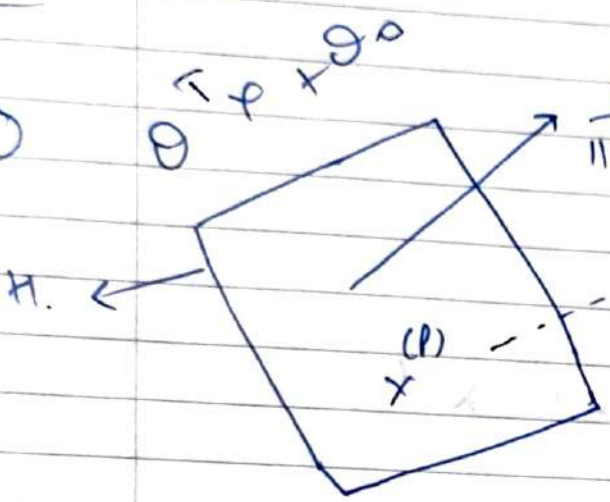
Fact: θ is normal vector to the surface $g_\theta(x) = 0$

$\frac{\theta}{\|\theta\|_2}$ is

(to make it unit normal vector dividing $\|\theta\|_2$)



(2)



$g_\theta(x^{(1)}) = 0$

$g_\theta(x^{(4)}) = 0$

* Pick a point $x^{(p)}$ on H.

* $x^{(p)}$ is the closed point (orthogonal projection) to $x^{(0)}$ (shortest distance)

* $x^{(0)} - x^{(p)}$ is the normal direction. So for some scaler $\beta > 0$

$$x^{(0)} - x^{(p)} = \beta \frac{\theta}{\|\theta\|_2}$$

→ As $x^{(p)}$ lies of H i.e $g_\theta(x^{(p)}) = \theta^T x^{(p)} + \theta_0 = 0$

→ $g_\theta(x^{(0)}) = \theta^T x^{(0)} + \theta_0$

$= \theta^T \left[x^{(p)} + \beta \frac{\theta}{\|\theta\|_2} \right] + \theta_0$

$$= \theta^T x^{(i)} + \theta_0 + \beta \frac{\theta^T \theta}{\|\theta\|_2}$$

$$g_\theta(x^{(i)}) = g(x^{(p)}) + \beta \frac{\|\theta\|^2}{\|\theta\|_2} = 0 + \beta \|\theta\|_2$$

$$\beta = \frac{g_\theta(x^{(i)})}{\|\theta\|_2} \quad \text{proved.}$$

So the closest point $x^{(p)}$ is

$$x^{(p)} = x^{(i)} - \beta \frac{\theta}{\|\theta\|_2}$$

$$x^{(p)} = x^{(i)} - \frac{g_\theta(x^{(i)})}{\|\theta\|_2} \cdot \frac{\theta}{\|\theta\|_2}$$

distance ⟷ normal vector

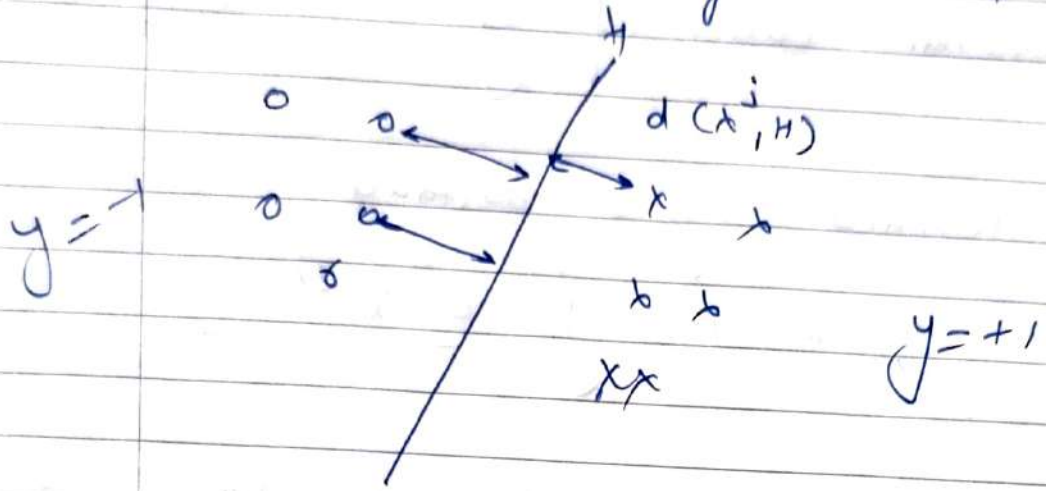
Unsigned Distance

(*) We define the distance b/w a data point $x^{(i)}$ & a separating hyperplane

$$d(x^{(i)}, H) = \frac{|g_\theta(x^{(i)})|}{\|\theta\|_2} = \frac{|\theta^T x^{(i)} + \theta_0|}{\|\theta\|_2}$$

$$= \frac{|\theta^T x^{(i)} + \theta_0|}{\|\theta\|_2}$$

$d(x^{(i)}, H)$ is unsigned distance



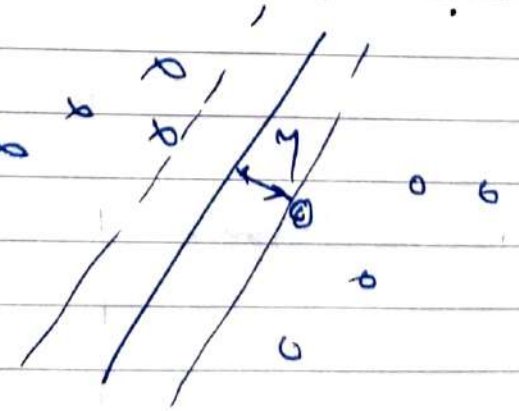
Margin (*) smallest distance among all the unsigned distances

$$\gamma = \min_{j=1,2,\dots m} d(x^{(j)}, H)$$

(*) γ is a function of θ & θ_0 (H is function of θ, θ_0)

(*) $\gamma > 0$ & is zero when $x^{(j)}$ is on the boundary

absolute value



Signed distance

* $d(x^{(i)}, H)$ is unsigned distance
* So y does not tell whether the point $x^{(i)}$ is correctly classified or not
* Assume labels of $y^{(i)} = \{-1, 1\}$

* We defined signed distance

$$d_{signed}(x^{(i)}, H) = y^{(i)} \frac{\theta^T x^{(i)} + \theta_0}{\|\theta\|_2}$$

$$= \begin{cases} \geq 0 & \text{correctly classify } x^{(i)} \\ < 0 & \text{incorrectly classify } x^{(i)} \end{cases}$$

Recall perceptron loss

$$L = \max(0, -y^{(i)}[\theta^T x^{(i)} + \theta_0])$$

Unsigned	Signed
$d \geq 0$	$d_s < 0$
* It's just a distance	It is a distance plus whether on the correct side or not

max margin objective (91)

* Assume data set is linearly seperable

It means

$$y^{(j)}\left(\frac{\theta^T x^{(j)} + \theta_0}{\|\theta\|_2}\right) \geq \gamma \quad j = 1, 2, \ldots n$$

condition ensure all data is correctly classified atleast greater than the margin γ /

→ Training samples are correctly classified

→ All samples are atleast γ from the boundary

* Max margin classifier is

$$\underset{\theta, \theta_0}{\text{maximize}} \ \gamma$$

subjected to $$y^{(j)}\left(\frac{\theta^T x^{(j)} + \theta_0}{\|\theta\|_2}\right) \geq \gamma$$

This constraint ~~problem~~ optimization problem is support vector machine problem.

Problem

$$\underset{\theta, \theta_0}{\text{maximize}} \ \gamma$$

subject to $y^{(i)}\left(\frac{\theta^T x^{(i)} + \theta_0}{\|\theta\|_2}\right) \geq \gamma$

* This optimization problem is not easy
* γ depends on (θ, θ_0)
* The term $\frac{1}{\|\theta\|_2}$ is non-linear

$\sqrt{\theta_1 + \theta_2 + \theta_3}$

Convert it into better optimization problem

scaling

(a) Let $x^{(m)}$ be point closest to the boundary

(b) Unsigned distance $\tilde{\gamma} = |\theta^T x^{(m)} + \theta_0|$

(c) $\gamma = \frac{|\theta^T x^{(m)} + \theta_0|}{\|\theta\|_2} = \frac{\tilde{\gamma}}{\|\theta\|_2}$

$$\begin{cases} \text{maximize} & \gamma \\ \theta, \theta_0 & \\ \text{subject} & y^{(i)}\left(\frac{\theta^T x^{(i)} + \theta_0}{\|\theta\|_2}\right) \geq \gamma \end{cases}$$

$$\max_{\theta, \theta_0} \frac{\tilde{\gamma}}{\|\theta\|_2}$$

$$\text{subject to } y^{(j)}\left(\frac{\theta^T x^{(j)} + \theta_0}{\|\theta\|_2}\right) \geq \frac{\tilde{\gamma}}{\|\theta\|_2}$$

$\frac{1}{\|\theta\|_2}$ goes to the objective function.

$$\max_{\theta, \theta_0} \frac{\tilde{\gamma}}{\|\theta\|_2}$$

$$\text{subject to } y^{(j)}(\theta^T x^{(j)} + \theta_0) \geq \tilde{\gamma}$$

$$j = 1, 2, \cdots$$

→ eliminate $\tilde{\gamma}$ by normalizing

$$\max_{\frac{\theta}{\tilde{\gamma}}, \frac{\theta_0}{\tilde{\gamma}}} \frac{1}{\left\|\frac{\theta}{\tilde{\gamma}}\right\|_2}$$

$$\text{subject to } y^{(j)}\left(\left(\frac{\theta}{\tilde{\gamma}}\right)^T x^{(j)} + \left(\frac{\theta_0}{\tilde{\gamma}}\right)\right) \geq 1$$

$$j = 1, 2, 3 \cdots m$$

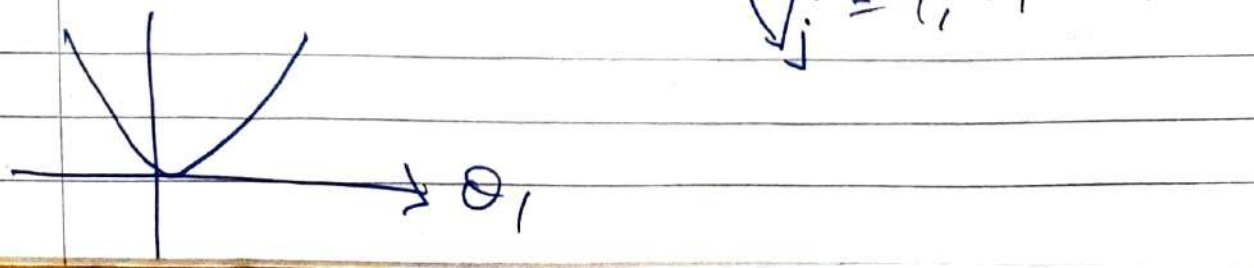
we can define $\theta \leftarrow \frac{\theta}{\tilde{\gamma}}$, $\theta_0 \leftarrow \frac{\theta_0}{\tilde{\gamma}}$

$$\begin{cases} \underset{\theta, \theta_0}{\text{maximize}} & \frac{1}{\|\theta\|_2} \\ \text{subject to} & y^{(j)}(\theta^T x^{(j)} + \theta_0) \geq 1 \\ & \forall j = 1, 2, \dots m \end{cases}$$

so $\tilde{\gamma}$ is eliminated

$$\begin{cases} \underset{\theta, \theta_0}{\text{maximize}} & \frac{1}{\|\theta\|_2} \\ \text{subject to} & y^{(j)}(\theta^T x^{(j)} + \theta_0) \geq 1 \\ & \forall j = 1, 2, \dots m \end{cases}$$

$$\begin{cases} \underset{\theta, \theta_0}{\text{minimize}} & \frac{1}{2}\|\theta\|^2 \quad \text{(ridge)} \\ \text{subject to} & y^{(j)}(\theta^T x^{(j)} + \theta_0) \geq 1 \\ & \forall j = 1, 2, \dots m \end{cases}$$



* This is a quadratic minimization with linear constraints
* It is convex
* Solution to this is called support vector machine

(Hard - SVM)

L-14 It is a quadratic programming problem

$$\min_u \frac{1}{2} u^T Q u + p^T u$$

$$s.t. \quad A u \geq c$$

soln. is $u^* = QP(Q, p, A, c)$. solved using QP solver

$$\begin{cases} \text{minimize}_{\theta, \theta_0} & \frac{1}{2} \|\theta\|^2 \\ \text{subject to} & y^{(j)}(\theta^T x^{(j)} + \theta_0) \geq 1 \end{cases}$$

$\forall j = 1, 2, \cdots m$

How to solve this

Approach :- Using Lagrangian function

we want to solve the following Inequality Constrained optimization problem

$$\text{minimize}_{x \in R^n} \quad f(x)$$

subject to $g_i(x) \geq 0 \quad , \quad i = 1 \ldots m$

$h_i(x) = 0 \quad , \quad i = 1 \ldots p$

where f, g_i are differentiable and convex h_i is affine

$h(x) = a^T x + b$ (affine)

It require a Lagrangian function

$$L(x, \alpha, \beta) = f(x) - \sum_{i=1}^{m} \alpha_i g_i(x) - \sum_{j=1}^{k} \beta_j h_j(x)$$

Here $\alpha_i \in \mathbb{R}^m$, $\beta \in \mathbb{R}^k$ are called as lagrange multipliers or dual variables, x is called as primal variable. Here $\alpha_i \geq 0$ and no constraint on β_j

if (x^*, α^*, β^*) is the soln. to the constrained optimization then all following conditions should hold

Karush - Kuhn - Tucker (KKT conditions)

1. $\nabla_x L(x^*, \alpha^*, \beta^*) = 0$ $\left(\frac{\partial L}{\partial x}\right)$

(*) stationary condition

(*) The primal variable should be stationary

(ii) $g_i(x^*) \leq 0$ & $h_{ij}(x^*) = 0$ for all i, j

(*) Primal feasibility

(*) Ensure that constrained constraints are satisfied

(iii) $\alpha_i^* \geq 0$ for all i & j

(*) Dual feasibility

(*) Require α_i^* ≥ but no constraint on β_i^*

(iv) ~~$\alpha_i^* g_i(x^*)$~~ $\alpha_i^* g_i(x^*) = 0$ for all i & j

(*) complementary slackness

(*) either $\alpha_i^* = 0$ or $g_i(x^*) = 0$ (or both)

$\alpha_i^* g_i(x^*) = 0$

$\alpha_i^* \geq 0$

$g_i(x^*) = 0$

$g_i(x^*) = 0$

Lagrangian function is given as

$$L(\theta, \theta_0, \lambda) = \frac{1}{2} \|\theta\|_2^2 + \sum_{j=1}^{m} \lambda_j \left[1 - y^{(j)} (\theta^T x^{(j)} + \theta_0) \right]$$

solution happened at

$\nabla_{(\theta, \theta_0)} L(\theta, \theta_0, \lambda) = 0$ & $\nabla_\lambda L(\theta, \theta_0, \lambda) = 0$

primal variables (a) (b)

Let re-write the lagrangian function again,

$$L(\theta, \theta_0, \lambda) = \frac{1}{2}\|\theta\|_2^2 + \sum_{j=1}^{m} \lambda_j \left[1 - y^{(j)}(\theta^T x^{(j)} + \theta_0)\right]$$

≤ 0

complementary condition -

- If $\lambda_j > 0$, then $1 - y^{(j)}(\theta^T x^{(j)} + \theta_0) = 0$
- if $\lambda_j = 0$ then $\left[1 - y^{(j)}(\theta^T x^{(j)} + \theta_0)\right] < 0$

useful in picking up svm

so if we take $\nabla_\lambda L(\theta, \theta_0, \lambda) = 0$

$$\nabla_\lambda L(\theta, \theta_0, \lambda) = 0 + \nabla_\lambda \left[\sum_{j=1}^{m} \lambda_j \left[1 - y^{(j)}(\theta^T x^{(j)} + \theta_0)\right]\right]$$

$= 0$

only possible if

$$\sum_{j=1}^{m} \lambda_j \left[1 - y^{(j)}(\theta^T x^{(j)} + \theta_0)\right]$$

maximize or minimum

If no point begins it is linear in λ

$$\lambda_j \left[1 - y^{(j)} (\theta^T x^{(i)} + \theta_0)\right]$$

<0 ⟵ ⟶ >0

→ If it is unbounded minimum so to find $\nabla_\lambda(\cdot)$ we must maximize this argument

By doing this we are finding the maximum pt. in the dual space

→ We can partition the optimization problem into two steps

① Let λ^* be the maximizer

$$\lambda^* = \underset{\lambda \geq 0}{\operatorname{argmax}} \left\{ \sum_{j=1}^{n} \lambda_j \left[1 - y^{(j)} (\theta^T x^{(j)} + \theta_0)\right] \right\}$$

② Then primal problem is (substituting result of ① in ② Pagini)

$$\min_{\theta, \theta_0} L(w, w_0, \lambda^*) = \min_{\theta, \theta_0} \left\{ \frac{1}{2} \|\theta\|^2 + \max_{\lambda \geq 0} \left\{ \sum_{j=1}^{n} \lambda_j \left[1 - y^{(j)} (\theta^T x^{i} + \theta_0) \right] \right\} \right\}$$

$$\min_{\theta, \theta_0} L(w, w_0, \lambda^*) = \min_{\theta, \theta_0} \left\{ \max_{\lambda \geq 0} L(\theta, \theta_0, \lambda^*) \right\}$$

This is min-max problem

$$\min_{\theta, \theta_0} \max_{\lambda \geq 0} \{ L(\theta, \theta_0, \lambda) \}$$

→ min max problem

Support vector machine! Duality

Two Reasons to go (duality) dual space!

(1) Can see why it is called as support vector

second will be discussed later

* Our problem is quadratic programming bcz

* strong duality hold for quadratic programming

$$\min_{\theta, \theta_0} \max_{\lambda \geq 0} L(\theta, \theta_0, \lambda) = \max_{\lambda \geq 0} \min_{\theta, \theta_0} L(\theta, \theta_0, \lambda)$$

primal space ← → dual space

$$\hat{\theta}, \hat{\theta}_0 = \max_{\lambda \geq 0} \min_{\theta, \theta_0} L(\theta, \theta_0, \lambda)$$ ← Dual Prob

$$\begin{cases} \underset{\theta, \theta_0}{\text{minimize}} \ \frac{1}{2}\|\theta\|^2 \\ \text{subject to} \quad y^{(j)}(\theta^T x^{(j)} + \theta_0) \geq 1 \quad \forall j = 1, 2, \ldots m \end{cases}$$

Lagrangian can be written as

$$L(\theta, \theta_0, \lambda) = \frac{1}{2}\|\theta\|_2^2 + \sum_{j=1}^{m} \lambda_j \left[1 - y^{(j)}(\theta^T x^{(j)} + \theta_0)\right]$$

In Dual setting, we will minimize w.r.t θ & θ_0 first

$$\begin{cases} \nabla_\theta L(\theta, \theta_0, \lambda) = \theta - \sum_{j=1}^{m} \lambda_j y^{(j)} x^{(j)} = 0 \\ \nabla_{\theta_0} L(\theta, \theta_0, \lambda) = \sum_{j=1}^{m} \lambda_j y^{(j)} = 0 \end{cases}$$

then maximize

→ stationary condition i.e KKT condition 1st

Interesting observation

$$\theta = \sum_{j=1}^{m} \lambda_j y^{(j)} x^{(j)}$$

Hyperplane is written as a linear combination of Data points.

Here λ_j is either zero or greater than zero

(*) Comes from complementary conditions

(*) Either use data point or not use it

Lagrangian function

$$L(\theta, \theta_0, \lambda) = \frac{1}{2}\|\theta\|_2^2 + \sum_{j=1}^{m} \lambda_j \left[1 - y^{(j)}\left(\theta^T x^{(j)} + \theta_0\right)\right]$$

$$= \frac{1}{2}\left\| \sum_{j=1}^{m} \lambda_j y^{(j)} x^{(j)} \right\|_2^2 + \sum_{j=1}^{m} \lambda_j \left[1 - y^{(j)}\left\{\left(\sum_{i=1}^{m} \lambda_i y^{(i)} x^{(i)}\right)^T x^{(j)} + \theta_0\right\}\right]$$

$$- \sum_{i=1}^{m}\sum_{j=1}^{m} \lambda_i \lambda_j y^{(i)} y^{(j)} (x^{(i)})^T x^{(j)} - \sum_{j=1}^{m} \lambda_j y^{(j)} \theta_0$$

$\theta_0 \rightarrow$ stationary condition

$$L(\theta, \theta_0, \lambda) = -\frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m} \lambda_i \lambda_j y^{(i)} y^{(j)} (x^{(i)})^T x^{(j)} + \sum_{j=1}^{m} \lambda_j$$

Now we do max as min is already evaluated for dual setting

$$\max_{\lambda} \left(\min_{\theta, \theta_0} L(\theta, \theta_0, \lambda) \right)$$

So dual problem is

maximize $-\frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\lambda_i\lambda_j y^{(i)}y^{(j)}(x^{(i)})^T x^{(j)} + \sum_{j=1}^{m}\lambda_j$

subject to $\sum_{j=1}^{m}\lambda_j y^{(j)} = 0$

→ Solution to this will be λ^*

complementary condition: $\lambda_j^*(1-y^{(j)}(\theta^{*T}x^{(j)}+\theta_0^*)) = 0$

(*) If $\lambda_j^* > 0$ then $1 - y^{*(j)}(\theta^{*T}x^{(j)}+\theta_0^*) = 0 \Rightarrow$

for all the pts. lying on the boundary

$y^{(j)}(\theta^T x^{(j)} + \theta_0^*) = 1$

the value of $\lambda_j^* > 0$

* If $\lambda_j^* = 0$, then $[1 - y^{(j)}(\theta^{*T}x^{(j)} + \theta_0^*)] < 0$

for all points correctly classified i.e lying in the interior of the boundary

$\{y^{(j)}[\theta^{*T}x^{(j)} + \theta_0^*) > 1\}$

the value of $\lambda_j^* = 0$

We can define sum as,

$$w = [\, j \mid \lambda_j^* > 0 \,] \Rightarrow x^{(j)} \text{ is on the margin}$$

so the optimal wts: $\theta^* \sum_{j \in w} \lambda_j^* y^{(j)} x^{(j)}$

→ The optimal seperating hyperplane depends on the support vector (points) closest to the decision boundary. It is different from least square solution, which depend on all training data

$\lambda > 0 \quad y^{(j)}(\theta^T x + \theta_0) = 1$

L-15

$y^{(1)}(\theta^T x^{(j)} + \theta_0) > 1$

$y^{(1)} = (\theta^T x + \theta_0)$

$y^{(j)} = 1$

$y^{(j)} = 1$

No data in this

$\theta^T x^{(j)} + \theta_0 = -1$

or $y^{(j)} = -1.$

$\theta^T x^{(j)} + \theta_0 = 1$

or $y^{(j)} = 1$

$\theta^T x^{(j)} + \theta_0 = 0$

$y^{(j)} [\theta^T x^{(j)} + \theta_0] = 1$

What about θ_0^* ?

(1) Pick any support vector $x^+ \in$ +ve class & $x^- \in$ -ve class

then we have

$$\theta^{*T} x^+ + \theta_0 = 1 \quad \& \quad \theta^{*T} x^- + \theta_0 = -1$$

$$\Rightarrow \quad \theta_0^* = - \frac{\theta^{*T} (x^+ + x^-)}{2}$$

Support vector machine :- Duality

Reason to go to dual space

(2) Dual enable kernel trick.

$$\text{maximize} \quad -\frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \lambda_i \lambda_j y^{(i)} y^{(j)} (x^{(i)T} x^{(j)}) + \sum_{j=1}^{m} \lambda_j$$

$$\lambda \geq 0$$

$$\text{subject to} \quad \sum_{j=1}^{m} \lambda_j y^{(j)} = 0$$

In matrix-vector format

$$\max_{\lambda \geq 0} \quad -\frac{1}{2} \lambda^T P \lambda + 1^T \lambda$$

$$\text{s.t.} \quad y^T \lambda = 0$$

Here

$$P = \begin{bmatrix} y^{(1)}y^{(1)}(x^{(1)})^T x^{(1)} & \cdots & y^{(1)}y^{(m)}(x^{(1)})^T x^{(m)} \\ y^{(2)}y^{(1)}(x^{(2)})^T x^{(1)} & \cdots & y^{(2)}y^{(m)}(x^{(2)})^T x^{(m)} \\ \vdots & & \\ y^{(m)}y^{(1)}(x^{(m)})^T x^{(1)} & \cdots & y^{(m)}y^{(m)}(x^{(m)})^T x^{(m)} \end{bmatrix}$$

Any observation?

Kernel trick

$$\text{maximize} \quad \sum_{i=1}^{m} \lambda_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \lambda_i \lambda_j y^{(i)} y^{(j)} (x^{(i)})^T x^{(j)} + \sum_{j=1}^{n} \lambda_j$$

$$\lambda \geq 0$$

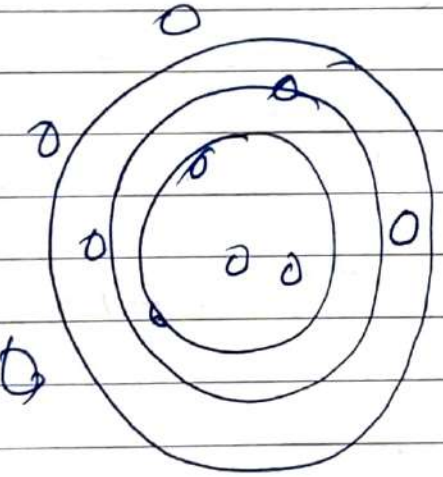
$$\text{subject to} \quad \sum_{j=1}^{m} \lambda_j y^{(j)} = 0$$

$$\text{maximize} \quad \sum_{i=1}^{m} \lambda_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \lambda_i \lambda_j y^{(i)} y^{(j)} \underbrace{\phi(x^{(i)})^T \phi(x^{(j)})}_{K(x^{(i)}, x^{(j)})} + \sum_{j=1}^{n} \lambda_j$$

$$\lambda \geq 0$$

$$\text{subject to} \quad \sum_{j=1}^{m} \lambda_j y^{(j)} = 0$$

$$P = \begin{bmatrix} y^{(1)}y^{(1)}k(x^{(1)},x^{(1)}) & \cdots & y^{(1)}y^{(m)}k(x^{(1)},x^{(m)}) \\ y^{(2)}y^{(1)}k(x^{(2)},x^{(1)}) & \cdots & y^{(2)}y^{(m)}k(x^{(2)},x^{(m)}) \\ \vdots & & \\ y^{(m)}y^{(1)}k(x^{(m)},x^{(1)}) & \cdots & y^{(m)}y^{(m)}k(x^{(m)},x^{(m)}) \end{bmatrix}$$

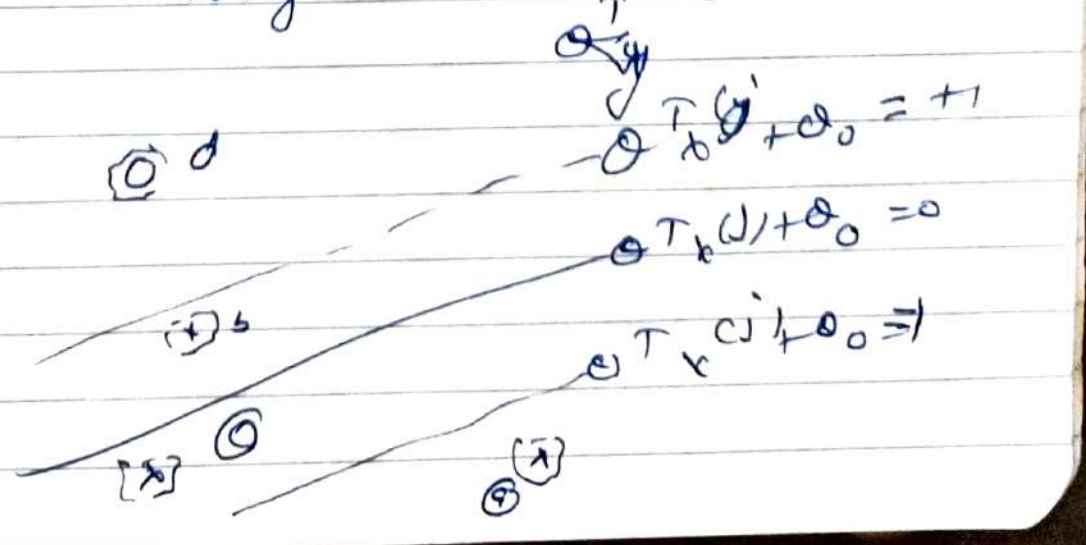


can we use linear svm here?

second order Hard svm (Kernel Boost)

Soft svm — Allow error into the margin

Hard svm :- not allow any error into the margin



for ③ $y^{(3)}(\theta^T x^{(3)} + \theta_0) = -1.5$

as $y^{(3)} = 1 \Rightarrow \theta^T x^{(3)} + \theta_0 = -1.5$

$\theta^T x^{(3)} + \theta_0 = 1 - (2.5)$ Eq1

for point ⑤ $y^{(5)}(\theta^T x^{(5)} + \theta_0) = 0.5$

$\Rightarrow \theta^T x^{(5)} + \theta_0 = 1 - (0.5)$ Eq2

for point ⓒ $y^{(c)}(\theta^T x^{(c)} + \theta_0) = -0.5$

$(\theta^T x^{(c)} + \theta_0) = 1 - (1.5)$ Eq3

for point ④ $y^{(d)}(\theta^T x^{(d)} + \theta_0) = 1.5$

as $y^{(d)} = -1$

$\Rightarrow \theta^T x^{(d)} + \theta_0 = -1.5 = 1 - (2.5)$ Eq4

→ changing hard margin $y^{(i)}(\theta^T x^{(i)} + \theta_0) \geq 1$ to soft margins (Hard SVM)

soft margin $y^{(i)}(\theta^T x^{(i)} + \theta_0) \geq 1 - \xi_i$, $\xi_i \geq 0$

if $\xi_i > 1$ → SVM

optimisation framework

$$\min_{\theta, \theta_0, \xi} \frac{1}{2}\|\theta\|_2^2 + C\|\xi\|^2$$

s.t $y^{(j)}(\theta^T x^{(j)} + \theta_0) \geq 1 - \xi_j$

$\xi_j \geq 0$ for $j = 1, 2, 3 \ldots n$.

If $C \uparrow$, $\xi \to 0$ → tendency to make mistakes on training data ↓ ie → overfitting

~~As~~ C ~~as~~ $C \downarrow$ ie, $\xi \uparrow$ i.e tendency to allow mistake to enter the margin ↑ so high bias (underfit)

Other variant:

$$\min_{\theta, \theta_0, \xi} \frac{1}{2}\|\theta\|_2^2 + C\|\xi\|_1$$

s.t $y^{(j)}(\theta^T x^{(j)} + \theta_0) \geq 1 - \xi_j$

$\xi_j \geq 0$ for $j = 1, 2, \ldots m$

→ Enforce ξ to be sparse

→ only few sample are allowed to lie in the margin

Connection with Perceptron Algorithm

In soft margin SVM, $\xi_i \geq 0$ & $y^{(j)}(\theta^T x^{(j)} + \theta_0) \geq 1 - \xi_i$

$\Rightarrow \xi_i \geq 0$ and $\xi_i \geq 1 - y^{(j)}(\theta^T x^{(j)} + \theta_0)$

These conditions can be combined together as

$$\xi_i \geq \max\left[0,\ 1 - y^{(j)}(\theta^T x^{(j)} + \theta_0)\right]$$

If SVM (soft) with 4 norm is used

$$L(\theta, \theta_0, \xi) = \frac{1}{2}\|\theta\|_2^2 + C\sum_{j=1}^{m} \xi_j$$

$$\geq \frac{1}{2}\|\theta\|_2^2 + C\sum_{j=1}^{m} \max\left[0,\ 1 - y^{(j)}(\theta^T x^{(j)} + \theta_0)\right]$$

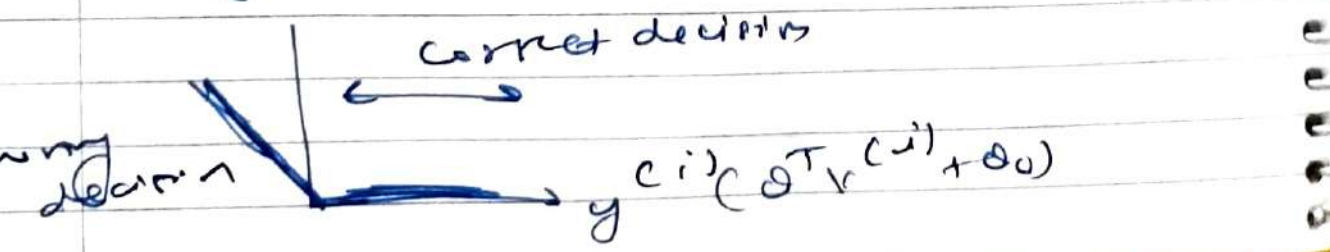
so the training loss becomes

$$L(\theta, \theta_0) = \sum_{j=1}^{m} \max\left[0,\ 1 - y^{(j)}(\theta^T x^{(j)} + \theta_0)\right] + \frac{\lambda}{2}\|\theta\|_2^2$$

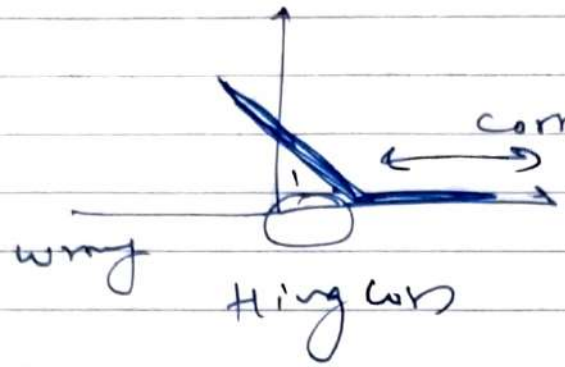
$$\lambda = \frac{1}{C}$$

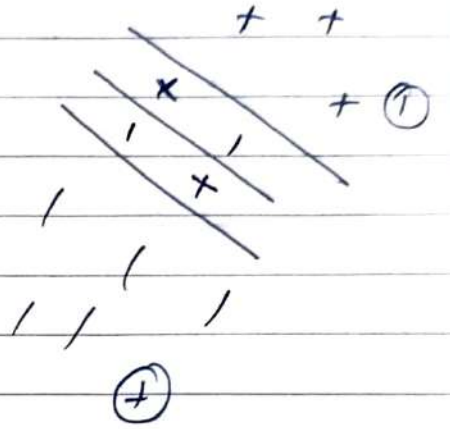
~~max(0, 1 - y^(j) (θ^T x^(j) + θ_0))~~

$\max(0, -y^{(j)}(\theta^T x^{(j)} + \theta_0))$



$$\max(0, 1 - y^{(i)}(\theta^T x^{(i)} + \theta_0))$$





L_2 loss

$$\sum_{i=1}^{m} \{y^{(i)} - \theta^T x^{(i)} + \theta_0)\}^2$$

As $y^{(i)} = \pm 1$ incase of classification

$$\sum_{i=1}^{m} \{1 - y^{(i)}(\theta^T x^{(i)} + \theta_0)\}^2$$

by taking $y^{(i)}$ outside possible due to squaring

Rewriting hinge loss

$$L(\theta, \theta_0) = \sum_{i=1}^{m} \max[0, 1 - y^{(i)}(\theta^T x^{(i)} + \theta_0)] + \frac{\lambda}{2}\|\theta\|^2$$

$$\lambda = \frac{1}{C}$$

If $\lambda \to 0 \Rightarrow C \to \infty$

So loss becomes

$$L(\theta, \theta_0) = \sum_{i=1}^{m} \max\left[0, 1 - y^{(i)}(\theta^T x^{(i)} + \theta_0)\right]$$

Perceptron loss $L = \sum_{i=1}^{m} \max\left[0, y^{(i)} x^{(i)} \theta\right]$

<u>Conclusion:</u> SVM (soft) L_1 norm penalty with $\lambda \to 0$ is perceptron (extreme situation)

In general SVM allow regularization to be added to perceptron loss thus regularizing the solution & bring the benefit of margin

solution & bring the ben