12.07.2022

# MACHINE LEARNING

## monotonic functions:

If we find $x$ which minimizes $Y = x$, it also minimizes $Y = x^2$. Therefore $x$ and $x^2$ are said to be monotonous.

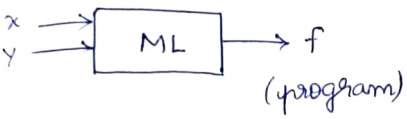
## machine learning:

model / data

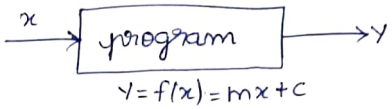
To frame model from given data.

We use regression to formulate the model.

ML




TP (traditional programming)




## ML LAB:

1) UCI, Kaggle } → ML repositories

* nature of $x$ and $Y$

| data types of $x$ | data types of $Y$ |
|---|---|
| continuous | continuous |
| discrete | discrete |
| multi-nomial | |
| ordinal / ordered | |
| nominal | |

* classification and regression

2) Download IRIS data set and house price prediction data set. Calculate mean, median, mode, standard deviation, covariance, skewness, kurtosis, coefficient of correlation

ML

* Machine learning is the study of algorithms that learn a function from data without being explicitly programmed.
* According to Tom Mitchell, ML is represented by the following triplet.

$$< T , P , E >$$

T — task; P — performance measure; E — experience

A ML program/algorithm is said to learn a task T from experience E, if it is able to improve the performance P in learning a task T over a period of time

TERMS OF ML:

h * hypothesis / function / model / concept - assumption

H * hypothesis space $H = \{h \mid h \text{ is } h(x)\}$ eg: $H = \{mx+c \mid m, c \in R\}$

* parameters - unknown values ($m$ & $c$ are parameters)
* hyperparameters - parameters used in training algo → eg: $f(x) = w_0 + w_1 x + w_2 x^2 + \dots + w_d x^d$ d-degree

ML ALGORITHMS:

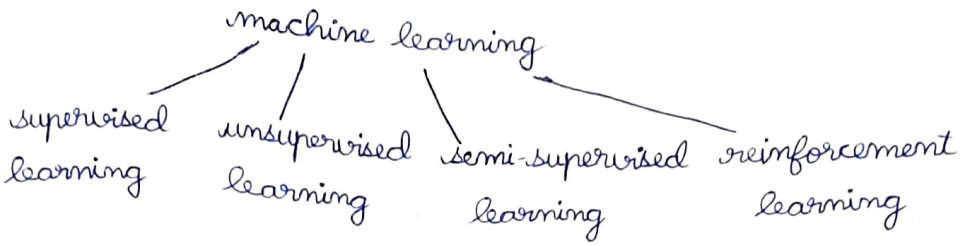
* linear regression
* polynomial regression
* Naive Bayes
* K - nearest neighbours
* support vector machines
* decision trees


- calculus
- linear algebra
  - eigen vector
  - eigen value
  - projection
  - vector space
  - norms
  - matrix factorization

→ building blocks

- probability & statistics
- optimization techniques

## CLASSIFICATION BASED ON DATA:



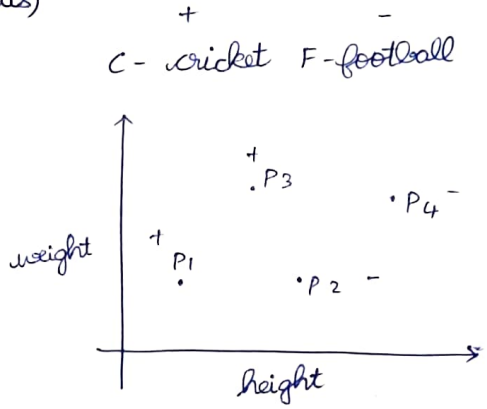

**Dataset:**

$$\mathcal{D} = \{ x_i, y_i \}_{i=1}^{n}$$

Dataset $\mathcal{D}$ consists of n observations, where $x_i$ represents input i and $y_i$ is its corresponding output i.

eg: 2 dimensional (∵ 2 inputs)

| | height | weight | type |
|---|---|---|---|
| $P_1$ | 5 | 60 | C |
| $P_2$ | 6 | 70 | F |
| $P_3$ | 6.1 | 80 | C |
| $P_4$ | 5.7 | 75 | F |



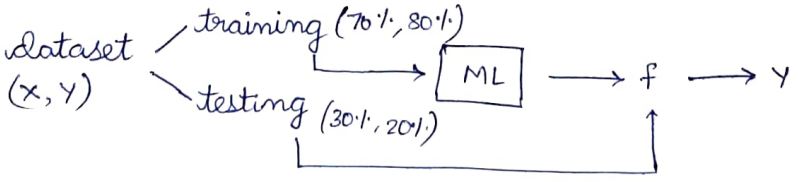

$x_i$ = input space / feature space / attributes

$y_i$ = output / class label / outcome

↑ discrete

$x_i \in R^2$

Dataset is considered as a matrix in which rows are considered as instances, datapoints, observations, examples and columns are features, attributes, characteristics, input vector.




## SUPERVISED LEARNING:

- classification (outcome is discrete)
- regression (outcome is continuous)

CLASSIFICATION ALGORITHMS :

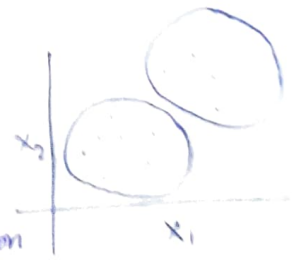
1. Naive Bayes
2. Logistic regression
3. Support vector machine (SVM)
4. k - nearest neighbours (KNN)
5. Linear Discriminant Analysis (LDA)
6. Quadratic Discriminant Analysis (QDA)
7. decision trees
8. Random forest

REGRESSION ALGORITHMS:

1. linear regression
2. polynomial regression
3. multiple regression
4. auto regression
5. KNN
6. SVM
7. decision trees

UNSUPERVISED LEARNING

→ only inputs → forms pattern
→ no output → no supervision



clustering

* In clustering, each group share common characteristics. Naming groups using domain knowledge becomes classification
* In dimension reduction, d dimensions is reduced to k dimensions $d \rightarrow k$
* Outlier analysis

ML LAB:

variance $\sigma^2 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n-1}$ → standard deviation = $\sqrt{\text{variance}}$

Boxplot - 2 variations

1. five no. summary (min, max, $Q_1$, $Q_2$ (median), $Q_3$) Q - quartiles
2. box - whisker plot

$Q_1$ → median of 1st half
$Q_3$ → median of 2nd half

outliers
* mild outliers
* strong outliers

SEMI-SUPERVISED LEARNING:

→ few inputs have output and the rest with no outputs.

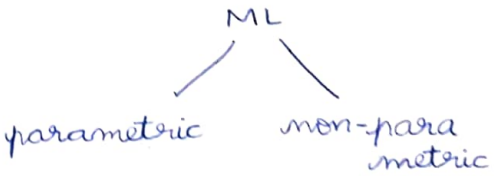
→ apply supervised adgorithms to data with outputs and build a confidence interval. Take data with high accuracy and apply ML algorithm to construct a model.

REINFORCEMENT LEARNING:

* Learning from sequence of moves. (penalty, rewards)
* No training set
* eg: robot learning

BASED ON MODEL:

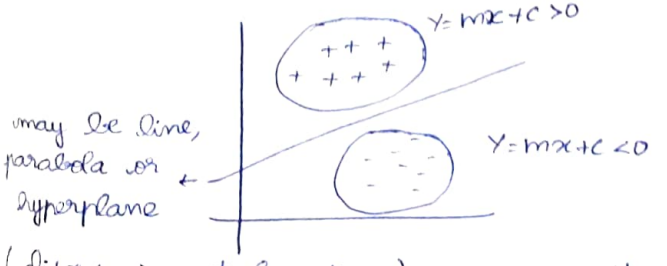

BASED ON PARAMETERS:




here)

DISCRIMINATIVE:

Divide data points in feature space according to its feature. (decision boundary)




(discriminant function)

ALGORITHMS:

* perceptron
* SVM
* logistic regression

constructs only 1 model

Discriminative finds conditional probability $P(Y|x)$ after analysing data points called posterior probability.

$Y \rightarrow$ output $\quad x \rightarrow$ input

## GENERATIVE:

* Based on learning data, we generate a model.
* It finds joint probability. $P(x, Y)$ * learns data space
* for every class we construct a model

## ALGORITHMS:

Naive Bayes, Hidden Markov Model (HMM)

## PARAMETRIC:

constructing a model with fixed set of parameters irrespective of number of data. Values of parameter changes.

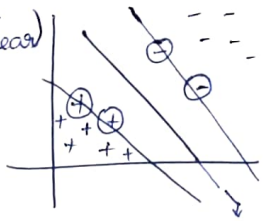
## ALGORITHMS:

* perceptron * SVM (linear) * linear & logistic regression

## NON-PARAMETRIC:

* no parameters
* number of parameters may vary

eg: SVM (non-linear)
KNN



* circled are called support vectors
* may vary when number of inputs vary

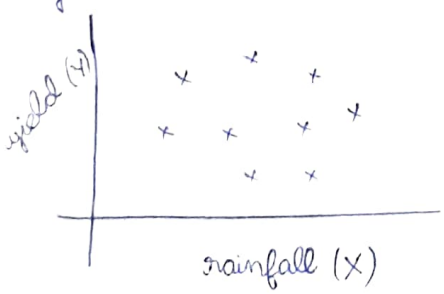
hyperplane * support vectors closest to hyperplan

## LINEAR REGRESSION: (supervised learning algorithm)

given $D = \{x_i, Y_i\}_{i=1}^{n}$ $\quad x_i \in R \quad Y_i \in R$

$f: x_i \rightarrow Y_i$ $\quad f \rightarrow$ true function $\hat{f} \rightarrow$ estimated function of $f$

objective: $\hat{f}: R \rightarrow R$




$x \rightarrow$ independent

$Y \rightarrow$ dependent on $x$

$Y = f(x) = mx + c$

linear relationship btw dependent and independent variables

P(Y/x)

r probability.

odel.

s data space
odel

neters
eter

n

pport

er of

to hyperplane

unction of f

endent
es

DATASET

| | x | Y |
|---|---|---|
| $P_1$ | 5 | 80 |
| $P_2$ | 6 | 90 |
| $P_3$ | 10 | 200 |
| $P_4$ | 20 | 400 |
| $P_5$ | 4 | 40 |
| $P_6$ | 7 | 75 |

$\boxed{f = \beta_0 + \beta_1 x}$

$\beta_0$ and $\beta_1$ are population parameters

By linear regression,

$\boxed{\hat{f} = b_0 + b_1 x_1}$ $\quad b_0 \approx \beta_0 \quad b_1 \approx \beta_1$

$b_0$ is an estimator of $\beta_0$ and

$b_1$ is an estimator of $\beta_1$

$\hat{f}: \hat{y} = b_0 + b_1 x$

$Y = b_0 + b_1 x + e$

$Y = \hat{y} + e$

$\therefore$ error $= Y - \hat{y}$ squared error $(Y - \hat{y})^2$

sum of squared errors $SSE = \boxed{\sum_{i=1}^{n} (Y_i - \hat{Y}_i)^2}$

Find $b_0$ and $b_1$ which minimizes SSE. (Optimization pblm)

$$SSE = \sum_{i=1}^{n} (Y_i - (b_0 + b_1 x_i))^2$$

$$\frac{\partial}{\partial b_0} SSE = 0$$

$$\frac{\partial}{\partial b_0} \sum_{i=1}^{n} (Y_i - (b_0 + b_1 x_i))^2 = 0$$

$$\sum_{i=1}^{n} 2 (Y_i - (b_0 + b_1 x_i)) (-1) = 0$$

$$-2 \sum_{i=1}^{n} (Y_i - (b_0 + b_1 x_i)) = 0$$

here)

$$\sum_{i=1}^{n} Y_i - \sum_{i=1}^{n} b_0 - \sum_{i=1}^{n} b_1 x_i = 0$$

$$\sum_{i=1}^{n} Y_i = \sum_{i=1}^{n} b_0 + \sum_{i=1}^{n} b_1 x_i$$

$$\sum_{i=1}^{n} Y_i = n b_0 + b_1 \sum_{i=1}^{n} x_i \longrightarrow (1)$$

$$\frac{\partial}{\partial b_1} SSE = 0$$

$$\frac{\partial}{\partial b_1} \sum_{i=1}^{n} (Y_i - (b_0 + b_1 x_i))^2 = 0$$

$$\sum_{i=1}^{n} 2 (Y_i - (b_0 + b_1 x_i)) (-x_i) = 0$$

$$-2 \sum_{i=1}^{n} (y_i - (b_0 + b_1 x_i)) x_i = 0$$

$$\sum_{i=1}^{n} y_i x_i - \sum_{i=1}^{n} b_0 x_i + \sum_{i=1}^{n} b_1 x_i^2 = 0$$

$$\sum_{i=1}^{n} y_i x_i = \sum_{i=1}^{n} b_0 x_i + \sum_{i=1}^{n} b_1 x_i^2$$

$$\sum_{i=1}^{n} y_i x_i = b_0 \sum_{i=1}^{n} x_i + b_1 \sum_{i=1}^{n} x_i^2 \longrightarrow (2)$$

From (1) and (2)

$$\begin{bmatrix} \Sigma y_i \\ \Sigma y_i x_i \end{bmatrix} = \begin{bmatrix} n & \Sigma x_i \\ \Sigma x_i & \Sigma x_i^2 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \end{bmatrix}$$

$$X = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ 1 & x_3 \\ \vdots & \vdots \\ 1 & x_n \end{bmatrix}_{n \times 2} \quad X^T = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ x_1 & x_2 & x_3 & \cdots & x_n \end{bmatrix}_{2 \times n} \quad b = \begin{bmatrix} b_0 \\ b_1 \end{bmatrix} \quad Y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}_{n \times 1}$$

$$X^T X = \begin{bmatrix} n & \Sigma x_i \\ \Sigma x_i & \Sigma x_i^2 \end{bmatrix}_{2 \times 2} \qquad X^T Y = \begin{bmatrix} \Sigma y_i \\ \Sigma x_i y_i \end{bmatrix}_{2 \times 1}$$

$Xb$ gives $\hat{y}$ values
∴ we added column of 1s

$$X^T Y = (X^T X) B$$

parametric & discriminative

$$\boxed{B = (X^T X)^{-1} X^T Y}$$ (closed form solution)

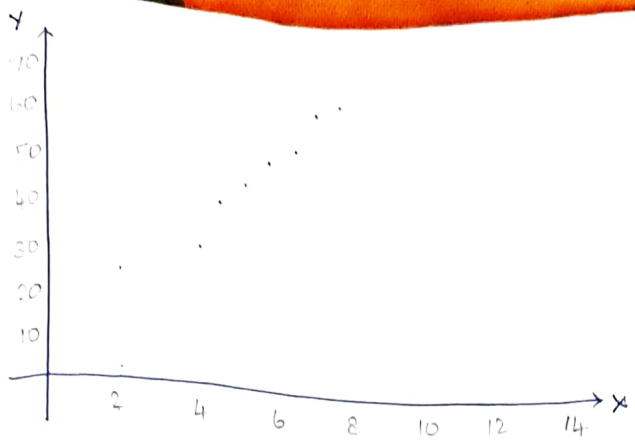
ML LAB:

1. X Soap 4 4.5 5 5.5 6 6.5 7

   Y Sud 33 42 45 51 53 61 62

$$X = \begin{bmatrix} 1 & 4 \\ 1 & 4.5 \\ 1 & 5 \\ 1 & 5.5 \\ 1 & 6 \\ 1 & 6.5 \\ 1 & 7 \end{bmatrix} \quad X^T = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 4 & 4.5 & 5 & 5.5 & 6 & 6.5 & 7 \end{bmatrix}$$

$$Y = \begin{bmatrix} 33 \\ 42 \\ 45 \\ 51 \\ 53 \\ 61 \\ 62 \end{bmatrix}$$




$X \propto Y$

X increases,
Y also increases

$$X^TX = \begin{bmatrix} 7 & 38.5 \\ 38.5 & 218.75 \end{bmatrix} \quad (X^TX)^{-1} = \begin{bmatrix} 4.454 & -0.785 \\ -0.785 & 0.143 \end{bmatrix} \quad X^TY = \begin{bmatrix} 347 \\ 1975 \end{bmatrix}$$

$$\begin{bmatrix} b_0 \\ b_1 \end{bmatrix} = (X^TX)^{-1}(X^TY) = \begin{bmatrix} -1.367 \\ 10.03 \end{bmatrix}$$

Predict sud value for soap value = 4.25.

$\hat{Y} = 37.696$

SSE = 19.96 SE = - 2.27

EVALUATION MEASURES :

total sum of squares (SST) = $\sum_{i=1}^{n} (y_i - \bar{y})^2$

sum of squared error (SSE) = $\sum_{i=1}^{n} (y_i - \hat{y}_i)^2$

regression sum of squares (SSR) = $\sum_{i=1}^{n} (\hat{y}_i - \bar{y})^2$

$0 <= SSR <= SST$

coeffecient of determination $r^2 = 1 - \frac{SSE}{SST}$ (or) $\frac{SSR}{SST}$

$SST = SSE + SSR$

$0 \leq r^2 \leq 1$

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

here)

2. Identify independent and dependent variables in house price data set. Draw scatter plot for every dep & indep feature and interpret the relationship between dep & indep feature and construct a model using linear regression for every dep & indep feature and evaluate models using different measures such as SSE, SST, SSR & coeff of determination & find out the most imp feature that influences dep variable. Find outliers if any.

A good model has greater SSR, $r^2$ and lesser SSE.

Soap Sud:

$$B = (X^T X)^{-1} X^T Y$$

$$X^T X = \underset{2\times 7}{\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 4 & 4.5 & 5 & 5.5 & 6 & 6.5 & 7 \end{bmatrix}} \underset{7\times 2}{\begin{bmatrix} 1 & 4 \\ 1 & 4.5 \\ 1 & 5 \\ 1 & 5.5 \\ 1 & 6 \\ 1 & 6.5 \\ 1 & 7 \end{bmatrix}}$$

$$X^T X = \begin{bmatrix} 7 & 38.5 \\ 38.5 & 218.75 \end{bmatrix} \qquad X^T Y = \begin{bmatrix} 347 \\ 1975 \end{bmatrix}$$

$$(X^T X)^{-1} = \begin{bmatrix} 4.464 & -0.786 \\ -0.786 & 0.143 \end{bmatrix}$$

※ don't round off

$$B = \begin{bmatrix} 4.464 & -0.785 \\ -0.785 & 0.143 \end{bmatrix} \begin{bmatrix} 347 \\ 1975 \end{bmatrix} = \begin{bmatrix} -3.342 \\ 9.683 \end{bmatrix} \quad \begin{bmatrix} -2.6786 \\ 9.5 \end{bmatrix}$$

$b_0 = -3.342 \quad b_1 = 9.683$

$$\hat{Y} = -3.342 + 9.683x$$

$\bar{Y} = 49.57$ $\quad \hat{Y} = -2.6786 + 9.5x$

| | | |
|---|---|---|
| $x_1 = 4$ | $\hat{Y}_1 = 35.39$ | 35.321 |
| $x_2 = 4.5$ | $\hat{Y}_2 = 40.2315$ | 40.071 |
| $x_3 = 5$ | $\hat{Y}_3 = 45.073$ | 44.821 |
| $x_4 = 5.5$ | $\hat{Y}_4 = 49.9145$ | 49.571 |
| $x_5 = 6$ | $\hat{Y}_5 = 54.756$ | 54.321 |
| $x_6 = 6.5$ | $\hat{Y}_6 = 59.5975$ | 59.071 |
| $x_7 = 7$ | $\hat{Y}_7 = 64.439$ | 63.821 |

$$SSE = (33 - 35.39)^2 + (42 - 40.2315)^2 + (45 - 45.073)^2 + (51 - 49.9145)^2 + (53 - 54.756)^2 + (61 - 59.5975)^2 + (62 - 64.439)^2$$

$$= 21.02$$

$SSR = 657.154$

$SST = 672.5992$

$r^2 = 0.9693$

$\rho = \frac{\text{covar}(x,y)}{\sigma_x \sigma_y}$ ⇒ coefficient of correlation ($r$)

$$\text{covar}(x,y) = \frac{\sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})}{n-1} \qquad \rho = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{n-1\, \sigma_x \sigma_y}$$

$$\sigma_x = \sqrt{\frac{\sum (x_i - \bar{x})^2}{n-1}} \qquad \sigma_y = \sqrt{\frac{\sum (y_i - \bar{y})^2}{n-1}} \qquad -1 \le r \le 1$$

$\rho = 0.9845$

eg: Take any [6] rows of iris data set. Construct a SLR model between sepal length and sepal width, petal length and petal width and calculate all measures.

| S.No | SEPAL LENGTH | SEPAL WIDTH | PETAL LENGTH | PETAL WIDTH |
|---|---|---|---|---|
| 1 | 5.1 | 3.5 | 1.4 | 0.2 |
| 2 | 4.9 | 3.0 | 1.4 | 0.2 |
| 3 | 4.7 | 3.2 | 1.3 | 0.2 |
| 4 | 4.6 | 3.1 | 1.5 | 0.2 |
| 5 | 5.0 | 3.6 | 1.4 | 0.2 |
| 6 | 5.4 | 3.9 | 1.7 | 0.4 |

X => sepal width $[3.5, 3.0, 3.2, 3.1, 3.6, 3.9]$

Y => sepal length $[5.1, 4.9, 4.7, 4.6, 5.0, 5.4]$

here)

$$X^T X = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 3.5 & 3.0 & 3.2 & 3.1 & 3.6 & 3.9 \end{bmatrix} \begin{bmatrix} 1 & 3.5 \\ 1 & 3.0 \\ 1 & 3.2 \\ 1 & 3.1 \\ 1 & 3.6 \\ 1 & 3.9 \end{bmatrix}$$

$$= \begin{bmatrix} 6 & 20.3 \\ 20.3 & 69.29 \end{bmatrix}$$

$$X^T Y = \begin{bmatrix} 29.7 \\ 100.91 \end{bmatrix} \qquad (X^T X)^{-1} = \begin{bmatrix} 19.6232 & -5.7507 \\ -5.7507 & 1.6997 \end{bmatrix}$$

$$B = \begin{bmatrix} 2.5059 \\ 0.72 \end{bmatrix} \qquad \hat{Y} = 2.5059 + 0.72x$$

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| X | 3.5 | 30 | 32 | 31 | 36 | 39 | |
| Y | 51 | 4.9 | 4.7 | 4.6 | 50 | 54 | $\bar{Y} = 4.95$ |
| $\hat{Y}$ | 5.0259 | 4.6659 | 4.8089 | 4.7379 | 5.0979 | 5.3139 | |

$SSE = 0.00549 + 0.054 + 0.0118 + 0.01901 + 0.0095 + 0.00741$

$SSE = 0.107$

$SST = 0.415$ $SSR = 0.306$ $r^2 = 0.737$ $r = 0.858$

---

X ⇒ petal width

Y ⇒ petal length

$$X^T X = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 0.2 & 0.2 & 0.2 & 0.2 & 0.2 & 0.4 \end{bmatrix} \begin{bmatrix} 1 & 0.2 \\ 1 & 0.2 \\ 1 & 0.2 \\ 1 & 0.2 \\ 1 & 0.2 \\ 1 & 0.4 \end{bmatrix}$$

$$= \begin{bmatrix} 6 & 1.4 \\ 1.4 & 0.36 \end{bmatrix}$$

$$X^T Y = \begin{bmatrix} 8.7 \\ 2.08 \end{bmatrix} \quad (X^T X)^{-1} = \begin{bmatrix} 1.8 & -7 \\ -7 & 30 \end{bmatrix}$$

$$B = \begin{bmatrix} 1.1 \\ 1.5 \end{bmatrix} \quad b_0 = 1.1 \quad b_1 = 1.5 \quad \hat{y} = 1.1 + 1.5x$$

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| X | 0.2 | 0.2 | 0.2 | 0.2 | 0.2 | 0.4 | |
| Y | 1.4 | 1.4 | 1.3 | 1.5 | 1.4 | 1.7 | $\bar{Y} = 1.45$ |
| $\hat{Y}$ | 1.4 | 1.4 | 1.4 | 1.4 | 1.4 | 1.7 | |

$SSE = 0.02$ $SST = 0.095$ $SSR = 0.075$ $r^2 = 0.7894$ $r = 0.8884$

$$B = (X^T X)^{-1} X^T Y$$

* Cannot use if $X^T X$ is not invertible ie) $|X^T X| = 0$
* when n becomes large, this $O(n^3)$ algorithm is ineffecient (2 matrix multiplication)

To overcome these drawbacks,

## GRADIENT DESCENT METHOD :

* iterative
* first order derivative
* unconstrained convex optimisation problem

## CONVEX SET:


$0 \le \lambda \le 1$

if $\lambda = 0$, $y$ | if $\lambda = 1$, $x$ | if $\lambda = 0.5$, midpoint

A set is a convex set if all points lie within the set. Any 2 points $x$ and $y \in$ convex set, then all points of $\lambda x + (1-\lambda)y \in$ convex set, then it is convex set.

eg: 

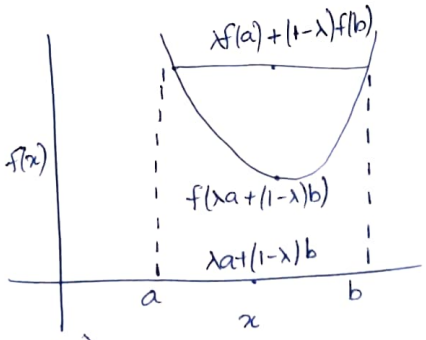
C NC

line, plane

intersection of 2 convex sets is always convex

union may or may not be convex

## CONVEX FUNCTION :

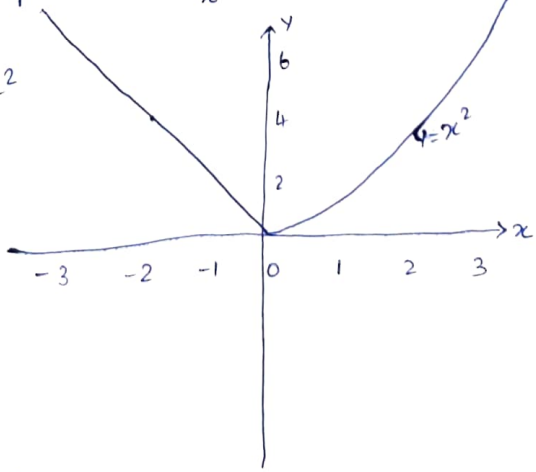



$f(\lambda a + (1-\lambda) b) \le \lambda f(a) + (1-\lambda) f(b)$

where $0 \le \lambda \le 1$

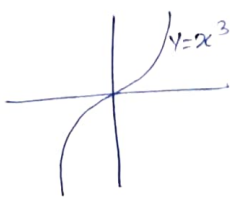
here)

$Y = x^2$




convex

$Y = x^3$




$Y = x^3$ if $x \ge 0$ then it is convex


not convex function

If $f(x)$ is convex then $-f(x)$ is concave and vice versa

All convex functions are $f''(x) \geq 0$ ※ and concave $f''(x) \leq 0$

Check whether following functions are convex:

1) $x \log x$ 3) $e^x \log x$

2) $-x \log x$ 4) $\log x$

1. $f(x) = x \log x$

$f'(x) = \log x + x \frac{1}{x} = \log x + 1$

$f''(x) = \frac{1}{x}$ convex for $x \geq 0$

2. $f(x) = -x \log x$

$f'(x) = -\log x + (-x)\frac{1}{x} = -\log x - 1$

$f''(x) = -\frac{1}{x}$ convex for $x \leq 0$

3. $f(x) = e^x \log x$

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

$f''(x) = e^x \left(\frac{-1}{x^2}\right) + \frac{1}{x} e^x + e^x \left(\frac{1}{x}\right) + \log x \, e^x$

$= e^x \left(\frac{1}{x} - \frac{1}{x^2} + \log x\right)$ not convex

4. $\log(x) \Rightarrow f(x)$

$f'(x) = \frac{1}{x}$ $f''(x) = \frac{-1}{x^2}$ not convex

Let $f(x, y, z)$ is a multivariate function

To check convexity

* find Hessian matrix H
* If H is positive definite / positive semidefinite, then f is convex

Jacobian matrix $J = \begin{bmatrix} \partial f/\partial x \\ \partial f/\partial y \\ \partial f/\partial z \end{bmatrix} = \begin{bmatrix} \frac{\partial^2 f}{\partial x^2} & \frac{\partial^2 f}{\partial x \partial y} & \frac{\partial^2 f}{\partial x \partial z} \\ \frac{\partial^2 f}{\partial y \partial x} & \frac{\partial^2 f}{\partial y \partial y} & \frac{\partial^2 f}{\partial y \partial z} \\ \frac{\partial^2 f}{\partial z \partial x} & \frac{\partial^2 f}{\partial z \partial y} & \frac{\partial^2 f}{\partial z^2} \end{bmatrix}$ (x, y, z)

H

All eigen values of H are $\geq 0 \rightarrow$ +ve semidefinite

$> 0 \rightarrow$ +ve definite

If matrix is symmetric, then it is +ve definite / +ve semi-definite. check for

$aHa^T \geq 0 \qquad aHa^T > 0$

H/W Check whether it is convex

$f(x_1, x_2, x_3) = (x_1 - x_2)^2 + 2x_3^2$

$J = \begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \\ \frac{\partial f}{\partial x_3} \end{bmatrix} = \begin{bmatrix} 2(x_1 - x_2)(1) \\ 2(x_1 - x_2)(-1) \\ 4x_3 \end{bmatrix} = \begin{bmatrix} 2x_1 - 2x_2 \\ 2x_2 - 2x_1 \\ 4x_3 \end{bmatrix}$

here)

$H = \begin{bmatrix} 2 & -2 & 0 \\ -2 & 2 & 0 \\ 0 & 0 & 4 \end{bmatrix}$

$Hx = \lambda x$

$(H - \lambda I)x = 0$

$|H - \lambda I| = 0$

H is symmetric

$$\begin{vmatrix} 2-\lambda & -2 & 0 \\ -2 & 2-\lambda & 0 \\ 0 & 0 & 4-\lambda \end{vmatrix} = 0$$

8
-4 -2

16
-4 -4

$$2-\lambda\left[(2-\lambda)(4-\lambda)\right] + 2\left[-2(4-\lambda)\right] = 0$$

$$(2-\lambda)\left[8-4\lambda-2\lambda+\lambda^2\right] + 2(-8+2\lambda) = 0$$

$$(2-\lambda)(\lambda^2-6\lambda+8) + (-16+4\lambda) = 0$$

$$2\lambda^2-12\lambda+16-\lambda^3+6\lambda^2-8\lambda-16+4\lambda = 0$$

$$-\lambda^3+8\lambda^2-16\lambda = 0$$

$$-\lambda(\lambda^2-8\lambda+16) = 0$$

$\boxed{\lambda=0}$ $\boxed{\lambda=4}$ $\boxed{\lambda=4}$ ∵ all eigen values $\geq 0$ +ve semi definite

It is a convex function

## CONVEX OPTIMISATION PROBLEM:

min $f(x)$ ⟶ convex function

s.t.c $g(x) \geq 0$ ⟶ convex set

* constrained COP
* unconstrained COP

} types

optimisation problem for LR:

Find $b_0$ and $b_1$ which minimises SSE.

$\hat{y}_i = b_0 + b_1 x_i$

$$\begin{bmatrix} b_0 & b_1 \end{bmatrix} \begin{bmatrix} 1 \\ x_i \end{bmatrix}$$

$B^T X$

$$L: \min \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 = \sum_{i=1}^{n} (y_i - B^T X_i)^2$$

$$\frac{\partial L}{\partial B} = \sum_{i=1}^{n} 2(y_i - B^T X_i)(-X_i)$$

$$\frac{\partial^2 L}{\partial B^2} = \sum_{i=1}^{n} -2X_i(-X_i) = \sum_{i=1}^{n} 2X_i^2 \geq 0$$

∴ SSE is a convex function

## LOSS FUNCTIONS: (all are convex)

1. SSE - linear reg, binary classification
2. MSE (mean squared error) - regression, binary classifi
3. negative log likelihood - binary classifi, logistic reg
4. hinge loss - SVM
5. cross entropy - multiple classification

## GRADIENT DESCENT METHOD

* unconstrained convex optimization problem
* first order partial derivative

Find $\theta$ which minimizes $f(\theta, x)$

$\theta$ is a parameter

gradient = $\begin{bmatrix} \partial f/\partial x_1 \\ \partial f/\partial x_2 \\ \partial f/\partial x_3 \end{bmatrix}$ (slope)

## ALGORITHM:

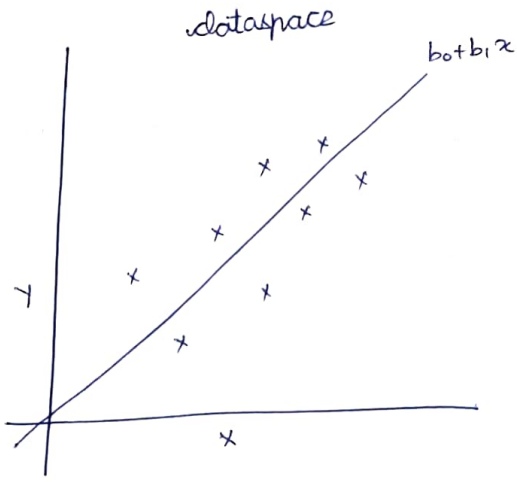
input $x$, $\eta$ (step size, learning rate)

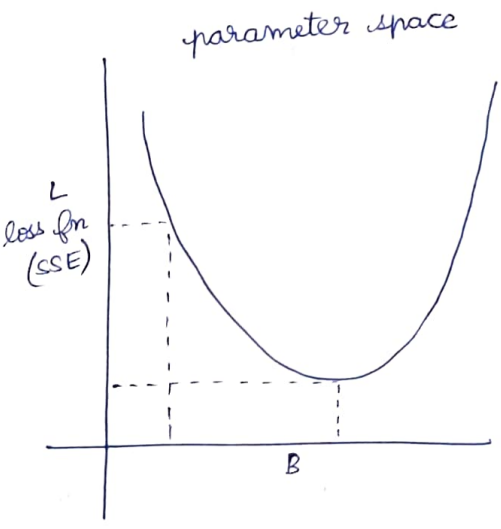
output $\theta^*$ $\quad \theta^* = \underset{\theta}{\text{argmin}}\ f(x, \theta)$

1. $\theta$ = initialize()
2. while (! convergence)

$\theta_{new} = \theta_{old} - \eta \frac{\partial f}{\partial \theta}$

$\theta_{old} = \theta_{new}$




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

$\sum (y_i - B^T x)^2$

$B^* = \underset{B}{\text{argmin}}\ SSE$

# LINEAR REGRESSION USING GRADIENT DESCENT:

input $\quad D = \{x_i, y_i\}_{i=1}^{n} \quad n$

output $\quad \vec{b} = \begin{bmatrix} b_0 \\ b_1 \end{bmatrix}$

1. $B$ = initialize()
2. while (! convergence) // $B_{new} = B_{old}$ (or) $\frac{\partial L}{\partial B} = 0$ — fixed number of epoch
   grad = [0,0]
   a. for each datapoint $(x_i, y_i)$

$$\hat{y_i} = b_0 + b_1 x_1$$

$$grad = grad + (y_i - \hat{y_i})(-x_i)$$

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

   c. $b_{old} = b_{new}$

PROBLEM :

$X = [4, 4.5, 5, 5.5, 6, 6.5, 7] \quad Y = [33, 42, 45, 51, 53, 61, 62]$

$\eta = 0.5 \quad B = [0.5, 0.5] \quad grad = [0, 0]$

IT-1

1. $\hat{y_i} = 0.5 + 0.5(4) = 2.5$

   $grad = [-30.5, -122]$

2. $\hat{y_i} = 0.5 + 0.5(4.5) = 2.75$

   $grad = [-30.5, -122] + [-39.25, -176.625] = [-69.75, -298.625]$

3. $\hat{y_i} = 0.5 + 0.5(5) = 3$

   $grad = [-69.75, -298.625] + [-42, -210] = [-111.75, -508.625]$

4. $\hat{y_i} = 0.5 + 0.5(5.5) = 3.25$

   $grad = [-111.75, -508.625] + [-47.75, -262.625] = [-159.5, -771.25]$

5. $\hat{y_i} = 0.5 + 0.5(6) = 3.5$

   $grad = [-159.5, -771.25] + [-49.5, -297] = [-209, -1068.25]$

6. $\hat{y_i} = 0.5 + 0.5(6.5) = 3.75$

   $grad = [-209, -1068.25] + [-57.25, -372.125] = [-266.25, -1440.375]$

7. $\hat{y_i} = 0.5 + 0.5(7) = 4$

   $grad = [-266.25, -1440.375] + [-58, -406] = [-324.25, -1846.375]$

$$B_{new} = B_{old} - \eta \, grad$$

$$= \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix} - 0.5 \begin{bmatrix} -324.25 \\ -1846.375 \end{bmatrix}$$

$$B_{new} = \begin{bmatrix} 162.625 \\ 923.6875 \end{bmatrix}$$

* one iteration is called <u>epoch</u>
* This method is called batch gradient descent / vanilla gradient descent method
* Since this is expensive, we go for stochastic gradient descent where we updates weights for each data point
* Since there will be more variation, we use mini batch gradient descent. If there 1000 inputs we divide it into 10 batches and weights are updated for every batch.

## ASSUMPTIONS OF LINEAR REGRESSION :

1. All observations are independant (iid)
2. Relationship between dependant & independant variables is linear.
3. Y is linear to parameters.
4. E(error) = 0 (mean)
5. For a $x_i$, error follow normal distribution.
6. No auto correlation between errors. (errors are indep here)
7. If Errors are heteroscedasticity, use weighted linear regressions. ∴ Errors are homoscedasticity. As x increases, error becomes constant.

## Linear regression is true function:

$$Y = (B_0 + B_1 x) + e \rightarrow \text{irreducible errors}$$

$$\hat{Y} = b_0 + b_1 x + e \rightarrow \text{SSE (reducible errors)}$$

## OUTLIER DETECTION TECHNIQUES :

1. Z Score (for converting data points to follow standard normal dist $\mu=0, \sigma=1$)

$$\begin{cases} \text{outlier} & \left|\frac{X-\mu}{\sigma}\right| \geq 3 \\ \text{no} & \text{otherwise} \end{cases}$$

2. Box plot

only for data points which follow normal dist

any points outside lower bound & upper bound

$\mu - 3\sigma \Rightarrow$ LB : $Q_1 - 1.5\,IQR$

$\mu + 3\sigma \Rightarrow$ UB : $Q_1 + 1.5\,IQR$

3. linear regression

construct model and choose the outliers (data points highly deviated from model). remove outliers and construct model again.

## POLYNOMIAL REGRESSION :

* instead of one independant feature $x$, we include powers of $x$.
* when data points itself follow polynomial function

$$Y = \beta_0 + \beta_1 x + \beta_2 x^2 + \ldots \beta_d x^d + \underbrace{\varepsilon}_{\text{irreducible errors}}$$

$$\hat{y} = b_0 + b_1 x + b_2 x^2 + \ldots b_d x^d$$

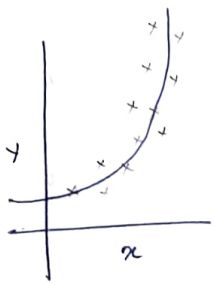
given

$$D = \{x_i, y_i\}_{i=1}^{n} \qquad x_i \in \mathbb{R} \quad y_i \in \mathbb{R}$$

$$f : x_i \rightarrow y_i \qquad (\beta_0 + \beta_1 x + \ldots \beta_d x^d) + e$$

objective

$$\hat{f} : \mathbb{R}^d \rightarrow \mathbb{R} \qquad b_0 + b_1 x + \ldots b_d x^d$$



optimization problem

Find $b = (b_0, b_1, \ldots b_d)$

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

where

$$\hat{y_i} = b_0 + b_1 x + b_2 x^2 + \ldots b_d x^d$$

$$L = \sum_{i=1}^{n} (y_i - (b_0 + b_1 x + b_2 x^2 + \ldots b_d x^d))^2$$

$$\frac{\partial L}{\partial b_0} = \Sigma\, 2\,(y_i - (b_0 + b_1 x + \ldots b_d x^d))(-1) = 0$$

$$\Sigma y_i = n b_0 + b_1 \Sigma x_i + b_2 \Sigma x_i^2 + \ldots + b_d \Sigma x_i^d$$

$$\frac{\partial L}{\partial b_1} = \Sigma\, 2\,(y_i - (b_0 + b_1 x + \ldots b_d x^d))(-x) = 0$$

$$\Sigma x_i y_i = b_0 \Sigma x_i + b_1 \Sigma x_i^2 + \ldots b_d \Sigma x_i^{d+1}$$

$$\frac{\partial L}{\partial b_2} = \Sigma\, 2\,(y_i - (b_0 + b_1 x + \ldots b_d x^d))(-x^2) = 0$$

$$\Sigma x_i^2 y_i = b_0 \Sigma x_i^2 + b_1 \Sigma x_i^3 + \ldots b_d \Sigma x_i^{d+2}$$

$$\vdots$$

$$\frac{\partial L}{\partial b_d} = \Sigma\, 2\,(y_i - (b_0 + b_1 x + \ldots b_d x^d))(-x^d) = 0$$

$$\Sigma x_i^d y_i = b_0 \Sigma x_i^d + b_1 \Sigma x_i^{d+1} \ldots + b_d \Sigma x_i^{2d}$$

$$Y = XB$$

where

$$Y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}_{n \times 1} \qquad X = \begin{bmatrix} 1 & x_1 & x_1^2 & \ldots & x_1^d \\ 1 & x_2 & x_2^2 & & x_2^d \\ 1 & x_3 & x_3^2 & & x_3^d \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_n & x_n^2 & \ldots & x_n^d \end{bmatrix}_{[n \times (d+1)]} \qquad B = \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ \vdots \\ b_d \end{bmatrix}_{(d+1) \times 1}$$

here)

$$X^T = \begin{bmatrix} 1 & 1 & 1 & 1 & \ldots & 1 \\ x_1 & x_2 & x_3 & x_4 & \ldots & x_n \\ x_1^2 & x_2^2 & x_3^2 & x_4^2 & \ldots & x_n^2 \\ \vdots & & & & & \vdots \\ x_1^d & x_2^d & x_3^d & x_4^d & \ldots & x_n^d \end{bmatrix}_{(d+1) \times n}$$

$$X^T X = \begin{bmatrix} n & \Sigma x_i & \Sigma x_i^2 & \ldots & \Sigma x_i^d \\ \Sigma x_i & \Sigma x_i^2 & \Sigma x_i^3 & \ldots & \Sigma x_i^{d+1} \\ \Sigma x_i^2 & \Sigma x_i^3 & \Sigma x_i^4 & \ldots & \Sigma x_i^{d+2} \\ \vdots & & & & \vdots \\ \Sigma x_i^d & \Sigma x_i^{d+1} & \Sigma x_i^{d+2} & \ldots & \Sigma x_i^{2d} \end{bmatrix}_{n \times n}$$

$$X^T Y = \begin{bmatrix} \Sigma y_i \\ \Sigma x_i y_i \\ \Sigma x_i^2 y_i \\ \vdots \\ \Sigma x_i^d y_i \end{bmatrix}$$

$$\boxed{B = (X^T X)^{-1} X^T Y}$$

eg:

| X | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| Y | 1 | 4 | 9 | 16 | 25 | 36 |

$\hat{y}_i = b_0 + b_1 x + b_2 x^2$

Construct second degree polynomial and find SSE.

Compare its SSE with linear regression model.

$$X = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 4 \\ 1 & 3 & 9 \\ 1 & 4 & 16 \\ 1 & 5 & 25 \\ 1 & 6 & 36 \end{bmatrix} \qquad X^T = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 & 5 & 6 \\ 1 & 4 & 9 & 16 & 25 & 36 \end{bmatrix}$$

$$X^T X = \begin{bmatrix} 6 & 21 & 91 \\ 21 & 91 & 441 \\ 91 & 441 & 2275 \end{bmatrix} \qquad Y = \begin{bmatrix} 1 \\ 4 \\ 9 \\ 16 \\ 25 \\ 36 \end{bmatrix}$$

$$X^T Y = \begin{bmatrix} 91 \\ 441 \\ 2275 \end{bmatrix}_{3\times1} \qquad (X^T X)^{-1} = \begin{bmatrix} 3.2 & -1.95 & 0.25 \\ -1.95 & 1.36964 & -0.1875 \\ 0.25 & -0.1875 & 0.02678 \end{bmatrix}_{3\times3}$$

$$B = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \qquad \therefore \boxed{\hat{y}_i = x^2} \qquad \boxed{SSE = 0}$$

By linear regression,

$\hat{y} = b_0 + b_1 x$

$$X = \begin{bmatrix} 1 & 1 \\ 1 & 2 \\ 1 & 3 \\ 1 & 4 \\ 1 & 5 \\ 1 & 6 \end{bmatrix} \qquad X^T = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 & 5 & 6 \end{bmatrix}$$

$$X^T X = \begin{bmatrix} 6 & 21 \\ 21 & 91 \end{bmatrix} \qquad Y = \begin{bmatrix} 1 \\ 4 \\ 9 \\ 16 \\ 25 \\ 36 \end{bmatrix}$$

$$(X^T X)^{-1} = \begin{bmatrix} 0.8666 & -0.2 \\ -0.2 & 0.057114 \end{bmatrix}$$

$$X^T Y = \begin{bmatrix} 91 \\ 441 \end{bmatrix}$$

$$B = (X^T X)^{-1} X^T Y = \begin{bmatrix} -9.333 \\ 7 \end{bmatrix}$$

$$\boxed{\hat{y} = -9.333 + 7x}$$

| $\hat{y}$ | -2.333 | 4.667 | 11.667 | 18.667 | 25.667 | |
|---|---|---|---|---|---|---|
| Y | 1 | 4 | 9 | 16 | 25 | |

$$SSE = \sum_{i=1}^{6} (y_i - \hat{y}_i)^2 = 37333334$$

$SSE_{LR} > SSE_{PR}$

ML LAB

Apply polynomial regression
every features and constru
A optimum model is a mo
error (SSE) or max $r^2$.

MULTIPLE REGRESSION :

Given $D = \{x_i, y_i\}_{i=1}^{n}$ $x_i \in$

$y_i = f(x_i)$

Objective

$\hat{f} : \mathbb{R}^d \to \mathbb{R}$

Hypothesis ( model is hypo

$Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots \beta_d$

$\hat{Y} = b_0 + b_1 x_1 + b_2 x_2 + \ldots b_d$
$w_0 + w_1 x_1 + w_2 x_2 + \ldots w_d$

$Y = XW + e$

$$Y = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_n \end{bmatrix}_{n\times1} \qquad W = \begin{bmatrix} w_0 \\ w_1 \\ w_2 \\ \vdots \\ w_d \end{bmatrix}_{(d+1)\times}$$

$$X^T Y = \begin{bmatrix} 91 \\ 441 \end{bmatrix}$$

$$B = (X^T X)^{-1} X^T Y = \begin{bmatrix} -9.333 \\ 7 \end{bmatrix}$$

$$\hat{Y} = -9.333 + 7x$$

| $\hat{Y}$ | -2.333 | 4.667 | 11.667 | 18.667 | 25.667 | 32.667 |
|---|---|---|---|---|---|---|
| Y | 1 | 4 | 9 | 16 | 25 | 36 |

$$SSE = \sum_{i=1}^{6} (Y_i - \hat{Y}_i)^2 = 37333334$$

$$SSE_{LR} > SSE_{PR}$$

ML LAB

Apply polynomial regression on iris data set between every features and construct an optimal model. A optimum model is a model which has lowest error (SSE) or max $r^2$.

MULTIPLE REGRESSION :

Given $D = \{X_i, Y_i\}_{i=1}^{n}$ $\quad X_i \in \mathbb{R}^d$ $\quad Y_i \in \mathbb{R}$

$Y_i = f(X_i)$

Objective

$\hat{f} : \mathbb{R}^d \rightarrow \mathbb{R}$

Hypothesis (model is hyperplane)

$Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots \beta_d x_d + \varepsilon \rightarrow$ random error

$\hat{Y} = b_0 + b_1 x_1 + b_2 x_2 + \ldots b_d x_d + e \rightarrow SSE$

$w_0 + w_1 x_1 + w_2 x_2 + \ldots w_d x_d + e$

$Y = XW + e$

$$Y = \begin{bmatrix} Y_1 \\ Y_2 \\ Y_3 \\ \vdots \\ Y_n \end{bmatrix}_{n \times 1} \quad W = \begin{bmatrix} w_0 \\ w_1 \\ w_2 \\ \vdots \\ w_d \end{bmatrix}_{(d+1) \times 1} \quad X = \begin{bmatrix} 1 & x_{11} & x_{12} & x_{13} & \ldots & x_{1d} \\ 1 & x_{21} & x_{22} & x_{23} & \ldots & x_{2d} \\ & & & & & \vdots \\ 1 & x_{n1} & x_{n2} & x_{n3} & \ldots & x_{nd} \end{bmatrix}_{n \times (d+1)}$$

## Optimisation problem

Find $W$

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

$$\min \; SSE(L) = \sum_{i=1}^{n} \left(y_i - (w_0 + w_1 x_{i1} + w_2 x_{i2} + \ldots + w_d x_{id})\right)^2$$

$$\frac{\partial L}{\partial w_0} = 2 \sum_{i=1}^{n} \left(y_i - (w_0 + w_1 x_{i1} + w_2 x_{i2} + \ldots + w_d x_{id})\right)(-1) = 0$$

$$\Rightarrow \Sigma y_i = n w_0 + w_1 \Sigma x_{i1} + \ldots + w_d \Sigma x_{id}$$

$$\frac{\partial L}{\partial w_1} \Rightarrow \Sigma y_i x_{i1} = w_0 \Sigma x_{i1} + w_1 \Sigma x_{i1}^2 + \ldots + w_d (\Sigma x_{id} x_{i1})$$

$$\vdots$$

$$\frac{\partial L}{\partial w_d} \Rightarrow \Sigma y_i x_{id} = w_0 \Sigma x_{id} + w_1 \Sigma x_{i1} x_{id} + \ldots + w_d \Sigma x_{id}^2$$

$$X^T = \begin{bmatrix} 1 & 1 & 1 & \ldots & 1 \\ x_{11} & x_{21} & x_{31} & \ldots & x_{n1} \\ \vdots & & & & \vdots \\ x_{1d} & x_{2d} & x_{3d} & \ldots & x_{nd} \end{bmatrix}_{(d+1) \times n}$$

$$X^T X = \begin{bmatrix} n & \Sigma x_{i1} & \Sigma x_{i2} & \ldots & \Sigma x_{id} \\ \Sigma x_{i1} & \Sigma x_{i1}^2 & \Sigma x_{i1} x_{i2} & \ldots & \Sigma x_{i1} x_{id} \\ \vdots & & & & \vdots \\ \Sigma x_{id} & \Sigma x_{id} x_{i1} & \Sigma x_{id} x_{i2} & \ldots & \Sigma x_{id}^2 \end{bmatrix}_{(d+1) \times (d+1)}$$

square matrix

$$X^T Y = \begin{bmatrix} \Sigma y_i \\ \Sigma y_i x_{i1} \\ \vdots \\ \Sigma y_i x_{id} \end{bmatrix}_{(d+1) \times 1}$$

$$\therefore X^T Y = (X^T X) W$$

$$\boxed{W = (X^T X)^{-1} X^T Y}$$

H/W

| CASES OF PRODUCT | 7 | 3 | 3 | 4 | 6 | 7 |
|---|---|---|---|---|---|---|
| DISTANCE (IN FT) | 560 | 220 | 340 | 80 | 150 | 330 |
| TIME (IN MINS) | 16:68 | 11:50 | 12:03 | 14:88 | 13:75 | 18:11 |

A soft drink bottler is analysing the vending machine serving routes in his distribution system. He is interested in predicting the time required by the driver to service the vending machines in an outlet.

$$\hat{y} = w_0 + w_1 x_1 + w_2 x_2$$

$$Y = \begin{bmatrix} 16.68 \\ 11.50 \\ 12.03 \\ 14.88 \\ 13.75 \\ 18.11 \end{bmatrix}_{6\times1} \qquad X = \begin{bmatrix} 1 & 7 & 560 \\ 1 & 3 & 220 \\ 1 & 3 & 340 \\ 1 & 4 & 80 \\ 1 & 6 & 150 \\ 1 & 7 & 330 \end{bmatrix}_{6\times3}$$

$d = 2$

$n = 6$

$$X^T = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 7 & 3 & 3 & 4 & 6 & 7 \\ 560 & 220 & 340 & 80 & 150 & 330 \end{bmatrix}_{3\times1}$$

$$W = (X^T X)^{-1} X^T Y$$

$$X^T X = \begin{bmatrix} 6 & 30 & 1680 \\ 30 & 168 & 9130 \\ 1680 & 9130 & 615400 \end{bmatrix}_{3\times3} \qquad X^T Y = \begin{bmatrix} 8695 \\ 456.14 \\ 25190.2 \end{bmatrix}_{3\times1}$$

$$W = \begin{bmatrix} 8.56558 \\ 1.19651 \\ -0.00020 \end{bmatrix}$$

## ASSUMPTION OF MULTIPLE REGRESSION:

no multicollinearity (no correlation between variables)

To eliminate correlated features (check for $dC_2$ times)

1. find correlation (keep threshold to remove 1 feature)

$$-1 \leq \sigma_{AB} = \frac{\text{covar}(A,B)}{\sigma_A \sigma_B} \leq +1$$

2. variance inflation factor (VIF)

$$VIF = \frac{1}{1-r^2}$$

$r^2$ - coeff of determination

$0 \leq r^2 \leq 1$

## MEASURES:

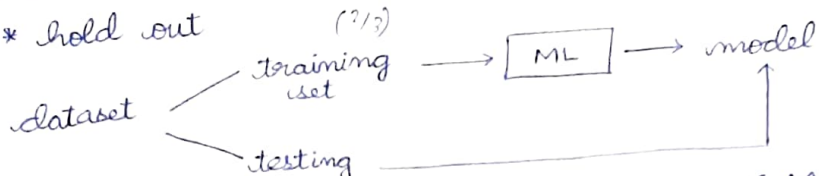
$$\text{adjusted } R^2 = 1 - \frac{(1-R^2)(n-1)}{(n-d-1)}$$

when d increases, $R^2$ decreases

use adjusted $R^2$ for multiple regression to avoid multicollinearity

## MODEL SELECTION:

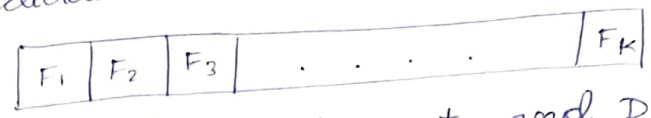
* hold out




find average of MSE to evaluate models.

* k fold cross validation

divide data set into k folds



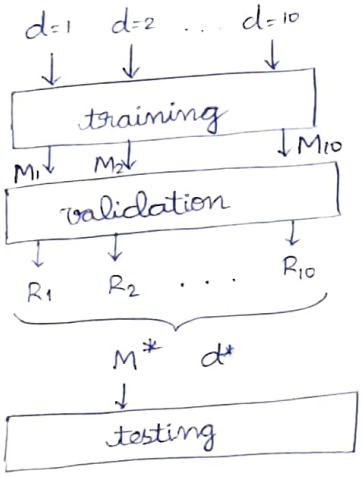

$i^{th}$ item used as testing set and D-Fi as training set

* leave one out

n fold cross validation (one fold one datapoint)

## FINDING OPTIMUM VALUE OF HYPERPARAMETER:

(3 way split)




M - model
R - coeff of correlation
d - degree

$$M^* = \underset{K}{\text{argmax}} \{R_{1K}\} \qquad 1 \le K \le 10$$

SSE is also called <u>training error</u>. If SSE / TE = 0 then the model passes through all data points.

| |
|---|
| bias - training error |
| variance - test error |

* If bias is low, model fits the training set well / complex model and if bias is high, model is simple / doesn't fit training set well.
* If variance is low, then deviation between testing error & training error is less.

low bias & low variance is the best model

| | low bias | high bias |
|---|---|---|
| low variance | ideal | simple |
| high variance | complex overfitting | simple underfitting |

* Overfitting and underfitting occur due to
  1) insufficient training set
  2) noise / outlier

* To increase size of dataset use
  1) augmentation (in image analysis, get data from all directions by tilting, rotating)
  2) smote analysis (for every data point create another synthetic data)

* Overfitting
  1) ridge regression / $L_2$ regularization
  2) lasso regression / $L_1$ regularization
  3) elastic net regression (combo of 1 & 2)

  } only for parametric problem

* regularization (applied only for parametric problems)

Find W

$$\min \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 + \text{regularization term} \leftrightarrow \|w\|^2$$

$L_1$ norm : $\text{abs}(w_0) + \text{abs}(w_1) + \ldots \text{abs}(w_d)$

$L_2$ norm : $\sqrt{w_0^2 + w_1^2 + \ldots w_d^2}$ (d dimensional circle / sphere)

$\|w\|$

$\|w\|$ is also convex.

∴ This optimisation problem can be solved by gradient descent method.

* Ridge regularization / $L_2$ reg

Find W

min SSE + $\lambda \cdot L_2$ norm of W

$$\Downarrow$$

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

$$\Downarrow$$

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

ML LAB:

Apply multiple regression on house price prediction by considering continuous features and evaluate the model using hold out method and calculate the average performance measures SSE, SST, SSR, adjusted $r^2$. Moreover, before constructing model, eliminate multi-collinearity if exists. Apply ridge and lasso regression and calculate measures. Compare the 3 models.

## Ridge regularization :

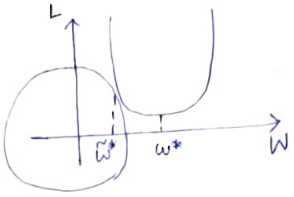
Find $W$

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

$$\Downarrow$$

Find $W$

$$\min \sum_{i=1}^{n} (y_i - W^T x_i)^2 \quad (x_i \text{ is a col vector})$$




$Y_{n\times 1}$

$X_{n\times d}$

$X^T_{d\times n}$

$W_{d\times 1}$

$W^T_{1\times d}$

S.t.c

$$w_1^2 + w_2^2 + w_3^2 + \ldots + w_d^2 = 1$$

$$\lambda \sum w_j^2 - \lambda$$

$$\Downarrow$$

Find $W$ $\min \sum_{i=1}^{n} (y_i - W^T x_i)^2 + \lambda \left( \sum_{j=1}^{d} w_j^2 - 1 \right)$

$\lambda \rightarrow$ lagrange multiplier

$$L = (Y - XW)^2 + \lambda W^T W \qquad L \rightarrow \text{scalar} \; \text{※}$$

$(Y - XW)_{n\times 1}$

$X^2 = X^T X$

$$L = (Y - XW)^T_{1\times n} (Y - XW)_{n\times 1} + \lambda W^T W$$

$$= Y^T Y - \underbrace{Y^T_{1\times n} X_{n\times d} W_{d\times 1}}_{\text{scalars}} - \underbrace{W^T_{1\times d} X^T_{d\times n} Y_{n\times 1}}_{\text{scalars}} + W^T X^T X W + \lambda W^T W$$

$$L = Y^T Y - 2 W^T X^T Y + W^T X^T X W + \lambda W^T W$$

$$\frac{\partial L}{\partial W} = -2 X^T Y + 2 W X^T X + 2 \lambda W = 0$$

$$X^T Y = W (X^T X + \lambda I)$$

$$\boxed{W = (X^T X + \lambda I)^{-1} X^T Y}$$

If $\lambda = 0$, ridge regularization becomes multiple regression.

If $\lambda \rightarrow \infty$, $W$ becomes close to 0. ∴ features are given least important and collinearity reduces

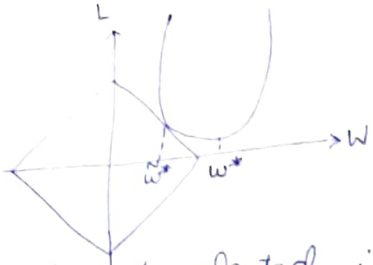
ASSUMPTIONS :

1. No bias (no $w_0$)
2. Data points are normalized (centered around the origin)

## Lasso Regularization :

Min
SSE + $L_1$ norm

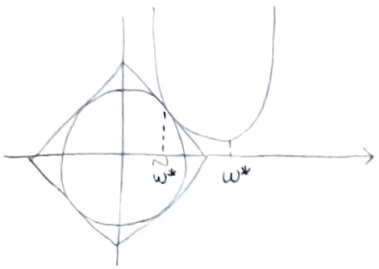
$$\sum (y_i - \hat{y_i})^2 + \lambda \sum_{j=1}^{d} |w_j|$$




When $\lambda \to \infty$, w becomes 0.
If w is 0 then particular feature is not selected. ∴ It can be used for feature selection technique.

## Elastic net:

combining both $L_1$ and $L_2$ norm




## CLASSIFICATION :

Given

$$\mathcal{D} = \{x_i, y_i\}_1^n$$

$x_i \in \mathbb{R}^d$ [discrete / continuous]

$y_i \in \{C_1, C_2\}$ (always discrete)

$f: x_i \to y_i$

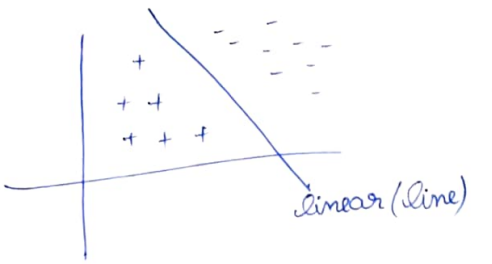
line – linear
plane – 2D
hyperplane – nD

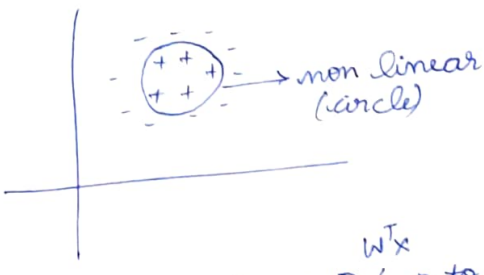
objective

$\hat{f}: \mathbb{R}^d \to \{C_1, C_2\}$

## Logistic regression

It is a linear classifier




$\mathcal{D} \Rightarrow$ linear regression $\Rightarrow w_0 + w_1 x_1 + w_2 x_2 + \cdots w_d x_d \Rightarrow \mathbb{R}$ ($w^T x$) ($-\infty$ to $\infty$)

$\Downarrow$

sigmoid function (converts $\hat{y}$ values from $-\infty$ to $\infty$ to 0 to 1 range)

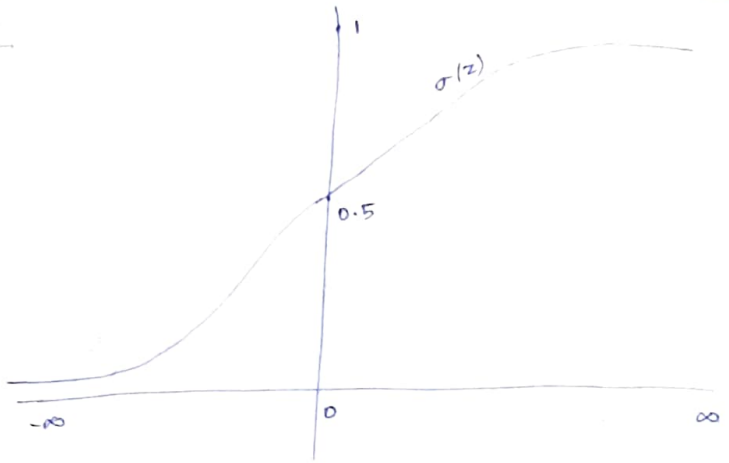
threshold

$$\hat{y} = \begin{cases} C_1 & \text{if } \sigma(w^T x) \geq 0.5 \\ C_2 & \text{o/w} \end{cases} \Leftarrow$$

$0 < \sigma(x) < 1$

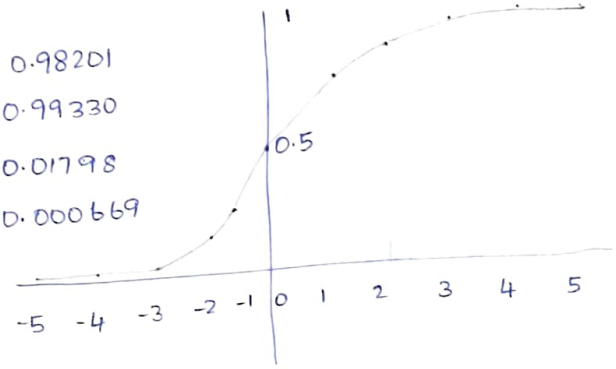
## SIGMOID FUNCTION:

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




Plot $\sigma(z)$ when z is in the range of -5 to +5.

| | | | |
|---|---|---|---|
| z = 0 | σ(z) = 0.5 | | |
| 1 | 0.73105 | 4 | 0.98201 |
| 2 | 0.88079 | 5 | 0.99330 |
| 3 | 0.95257 | -4 | 0.01798 |
| -1 | 0.2689 | -5 | 0.000669 |
| -2 | 0.11920 | | |
| -3 | 0.04742 | | |




## PROPERTIES: S-shaped function

* $0 < \sigma(z) < 1 \quad \forall z, -\infty \le z \le \infty$
* $\sigma(z) = 1/2$ for $z = 0$
* $z \to +\infty$ then $\sigma(z)$ is close to 1
* $z \to -\infty$ then $\sigma(z)$ is close to 0
* if $\sigma(z) = \frac{1}{1+e^{-z}}$ $\quad 1 - \sigma(z) = \frac{e^{-z}}{1+e^{-z}}$
* $\sigma'(z) = \frac{0+e^{-z}}{(1+e^{-z})^2} = \frac{e^{-z}}{(1+e^{-z})^2} = \sigma(z)\,(1-\sigma(z))$

Derivative of $\sigma(z)$ can be written in terms of $\sigma(z)$ itself.

In classification, SSE gives total number of misclassification. $\sum_{i=1}^{n} (y_i - \hat{y}_i)^2$ (incase of binary classification)

If y is a Bernoulli random variable,

$$f(y) = p^y (1-p)^{1-y}$$

if $y = 0 \quad f(y) = 1-p$

if $y = 1 \quad f(y) = p$

# LOGISTIC REGRESSION : (discriminative, parametric model)

Given

$$D = \{x_i, y_i\}_{i=1}^{N} \quad x_i \in \mathbb{R}^d \ (x_i \in N(\mu, \sigma)) \quad y_i \in \{C_1, C_2\}$$

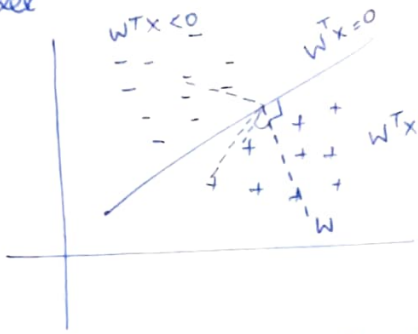
$$f : x_i \to y_i$$

Objective

$$\hat{f} : \mathbb{R}^d \to \{C_1, C_2\}$$

Assumptions:

1. $x_i$ is continuous $x_i \in N(\mu, \sigma)$
2. No or less multicollinearity between features.
3. Model $P(Y = C_k | X) = \frac{1}{1+e^{-w^T x}}$ ; $P(Y = C_2 | X) = 1 - \frac{1}{1+e^{-w^T x}} = \frac{e^{-w^T x}}{1+e^{-w^T x}}$
4. Model




Hyperplane $H = \{x \mid W^T X = 0\}$

If $W^T X = 0$ then they are Ir. (dot product of 2 vectors) (or orthogonal)

$$W = \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_d \end{bmatrix} \quad X = \begin{bmatrix} 1 \\ x_1 \\ \vdots \\ x_d \end{bmatrix}$$

$$W^T X = \begin{bmatrix} w_0 & w_1 & \cdots & w_d \end{bmatrix} \begin{bmatrix} 1 \\ x_1 \\ \vdots \\ x_d \end{bmatrix}$$

$$= w_0 + w_1 x_1 + \cdots w_d x_d$$

$$= w_0 + \sum_{j=1}^{d} w_j x_j$$

$$W \cdot X = \|W\| \|X\| \cos\theta$$

all data points in +ve region have acute angle with W ($<90°$)

all data points in -ve region have obtuse angle with W ($>90°$)

$$\hat{y} = \begin{cases} C_1 & W^T X > 0 \\ C_2 & W^T X < 0 \end{cases}$$

$$\hat{y} = \begin{cases} C_1 & \text{if } P(Y = C_1 | X) \geq P(Y = C_2 | X) \\ C_2 & \text{otherwise} \end{cases}$$

$$= \begin{cases} C_1 & \text{if } \frac{1}{1+e^{-W^T x}} \geq 0.5 \\ C_2 & \text{otherwise} \end{cases}$$

Find W

which correctly classifies all +ve & -ve (or)

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

$$\min \sum_{i=1}^{n} (y_i - \sigma(W^T x_i))^2$$

$$W_{new} = W_{old} - \eta \frac{\partial L}{\partial W}$$

$$\frac{\partial L}{\partial W} = -2 \sum_{i=1}^{n} (y_i - \sigma(W^T x_i)) (\sigma(W^T x_i)(1 - \sigma(W^T x_i))) x_i \quad \Big| \quad \hat{y}_i = \sigma(W^T x_i)$$

$$= -2 \sum (y_i - \hat{y}_i) \hat{y}_i (1 - \hat{y}_i) x_i$$

GRADIENT DESCENT:

1. $W_{old}$ = initialize ()
2. while !! convergence) {

   for each $(x_i, y_i) \in D$

   $$\hat{y}_i = \begin{cases} c_1 & \text{if } \sigma(W^T x_i) \geq 0.5 \\ c_2 & \text{otherwise} \end{cases}$$

   if $y_i != \hat{y}_i$

   $$W_{new} = W_{old} + \eta (y_i - \hat{y}_i) \hat{y}_i (1 - \hat{y}_i) x_i$$

}

TEST-1

1.a.i. $L = y_i \log \hat{y}_i \qquad \hat{y}_i = W^T X$

$$L = y_i \log W^T X$$

$$L' = \frac{y_i}{W^T X}(x) = \frac{y_i}{W^T} \qquad L'' = \frac{-y_i}{(W^T)^2} \leq 0 \quad \therefore \text{concave}$$

* epoch - weight updation for every data items after each iteration of data item's grad calculation

2.a. SE (sum of residuals / errors) = 0 = $\Sigma (y_i - \hat{y}_i)$ ✗

for linear regression

| $X_1$ | $X_2$ | Y |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

$C_1 \Rightarrow 1 \quad C_1 \Rightarrow 0$

Construct a logistic regression model for the given data using SGD method with $\eta = 0.5$, $W_0 = 0.5$, $W_1 = -0.5$, $W_2 = 0.5$.

| | $X_1$ | $X_2$ | $W_0$ | $W_1$ | $W_2$ | Y | $W^T X$ | $\sigma(W^T X)$ | $\hat{Y}$ |
|---|---|---|---|---|---|---|---|---|---|
| epoch1 | 0 | 0 | 0.5 | -0.5 | 0.5 | 0 | 0.5 | 0.622459 | $C_1$ |
| | 0 | 1 | 0.42 | -0.5 | 0.5 | 1 | 0.92 | 0.715 | $C_1$ |
| | 1 | 0 | 0.42 | -0.5 | 0.5 | 1 | -0.08 | 0.48 | $C_2$ |
| | 1 | 1 | 0.48 | -0.44 | 0.5 | 1 | 0.54 | 0.63 | $C_1$ |
| epoch2 | 0 | 0 | 0.48 | -0.44 | 0.5 | 0 | 0.48 | 0.62 | $C_1$ |
| | 0 | 1 | 0.41 | -0.44 | 0.5 | 1 | 0.91 | 0.71 | $C_1$ |
| | 1 | 0 | 0.41 | -0.44 | 0.5 | 1 | -0.03 | 0.49 | $C_2$ |
| | 1 | 1 | 0.47 | -0.38 | 0.5 | 1 | 0.59 | 0.64 | $C_1$ |

if $Y$ and $\hat{Y}$ are not equal update W values

$$W_{new} = \begin{bmatrix} 0.5 \\ -0.5 \\ 0.5 \end{bmatrix} + (0.5)(-0.62)(0.62)(0.38) \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

$$= \begin{bmatrix} 0.5 \\ -0.5 \\ 0.5 \end{bmatrix} + \begin{bmatrix} -0.073 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0.427 \\ -0.5 \\ 0.5 \end{bmatrix}$$

$$W_{new} = \begin{bmatrix} 0.42 \\ -0.5 \\ 0.5 \end{bmatrix} + (0.5)(0.52)(0.48)(0.52) \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}$$

$$= \begin{bmatrix} 0.42 \\ -0.5 \\ 0.5 \end{bmatrix} + \begin{bmatrix} 0.06 \\ 0.06 \\ 0 \end{bmatrix} = \begin{bmatrix} 0.48 \\ -0.44 \\ 0.5 \end{bmatrix}$$

$$W_{new} = \begin{bmatrix} 0.48 \\ -0.44 \\ 0.5 \end{bmatrix} + (0.5)(-0.62)(0.62)(0.38) \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

$$= \begin{bmatrix} 0.407 \\ -0.44 \\ 0.5 \end{bmatrix}$$

$$W_{new} = \begin{bmatrix} 0.41 \\ -0.44 \\ 0.5 \end{bmatrix} + (0.5)(0.51)(0.49)(0.51) \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}$$

$$= \begin{bmatrix} 0.41 \\ -0.44 \\ 0.5 \end{bmatrix} + (0.06) \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0.47 \\ -0.38 \\ 0.5 \end{bmatrix}$$

PROVE

Logistic regression is a linear model.

PROOF:

$$P(Y=1|X) = \frac{1}{1+e^{-W^TX}}$$

$$P(Y=0|X) = 1 - P(Y=1|X) = \frac{e^{-W^TX}}{1+e^{-W^TX}}$$

$$\text{odds of success} = \frac{P(Y=1|X)}{P(Y=0|X)}$$

$$= \frac{1}{e^{-W^TX}} = e^{W^TX}$$

$$\text{log odds of success} = \log_e e^{W^TX}$$

↓ (logit model)

$$= W^TX \text{ is linear}$$

Logistic reg → sigmoid function is called logistic function

P(B) = 22/38 P(G) = 16/38

odds probability : total = 38

# boys = 22 # girls = 16

odds of girl = $\frac{16}{22} = \frac{8}{11} = \frac{P(G)}{P(B)}$

odds of boys = $\frac{22}{16} = \frac{11}{8} = \frac{P(B)}{P(G)}$

odd = $\frac{\text{probability (success)}}{\text{probability (failure)}}$

* Find the odds of each category of flower in IRIS data set:

odd of each class = $\frac{1}{2}$ ($\because$ each class 50, total 150, $P(\text{class}) = \frac{1}{3}$)

EVALUATION METRICS / MEASURES:

| | |
|---|---|
| accuracy | true positive rate (TPR) |
| precision | false positive rate (FPR) |
| recall | receiver operating characteristic curve (ROC) |
| F-measure | area under the ROC curve (AUC) |

## confusion matrix

| actual \ predicted | + | − | |
|---|---|---|---|
| + | true positive | false negative | $P = TP + FN$ |
| − | false positive | true negative | $N = FP + TN$ |

$\hat{P} = TP + FP \quad \hat{N} = FN + TN$

1) accuracy $= \frac{TP+TN}{TP+TN+FP+FN} = \frac{TP+TN}{n}$ $\quad n$ - total no of datapoints

$= \frac{TP+TN}{P+N} = \frac{TP+TN}{\hat{P}+\hat{N}}$

eg: $n = 100 \quad TP = 35 \quad TN = 25$

50+ 50−

balanced dataset

| | | |
|---|---|---|
| 35 | 15 | 50 |
| 25 | 25 | 50 |
| 60 | 40 | |

accuracy $= \frac{60}{100} = 0.60$

eg: $n = 100$

90− 10+

skewed dataset / imbalanced

$TP = 0 \quad TN = 90$

| | |
|---|---|
| 0 | 10 |
| 0 | 90 |

accuracy $= \frac{90}{100} = 0.90$

since we are interested in +ve, accuracy is not a good measure for skewed dataset (it didn't predict single +ve correct)

2) precision $= \frac{TP}{\hat{P}} \Big/ \frac{TN}{\hat{N}}$ (no of +ve predictions w.r.t total +ve predictions)

3) recall $= \frac{TP}{P} \Big/ \frac{TN}{N}$ (no. of +ve predictions w.r.t total +ve in actual dataset)

4) F - measure = harmonic mean of precision and recall

$= \frac{2}{\frac{1}{P}+\frac{1}{R}} = \frac{2}{\frac{P+R}{PR}} = \frac{2PR}{P+R}$

(arithmetic mean is not used, since it is varies with extreme)

5) TPR $= \frac{TP}{P}$ (increase TPR)

6) FPR $= \frac{FP}{N}$ (decrease FPR)

} both are disjoint

7) TNR $= \frac{TN}{N}$

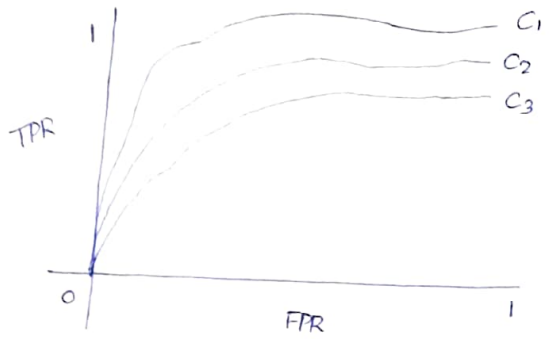
8) FNR $= \frac{FN}{P}$

TPR : recall

(Calculate all measures)

# RECEIVER OPERATING CHARACTERISTIC CURVE : (ROC CURVE)

TPR Vs FPR




$C_1 > C_2 > C_3$

TPR and FPR are chosen because they are disjoint TP w.r.t P and FP w.r.t N.

| TPR $\left(\frac{TP}{P} = \frac{TP}{TP+FN}\right)$ | | FPR $\left(\frac{FP}{N} = \frac{FP}{FP+TN}\right)$ | | INFERENCE |
|---|---|---|---|---|
| 0 | TP=0, P=FN | 0 | FP=0, N=TN | all data → N<br>biased to N class |
| 0 | TP=0, P=FN | 1 | FP=1, TN=0, N=FP | all +ve → N<br>all -ve → P |
| 1 | TP=1, P=TP, FN=0 | 0 | FP=0 TN=N | all datapoints correctly classified |
| 1 | TP=1, FN=0, P=TP | 1 | FP=1, TN=0, N=FP | all data → P<br>biased to P class |

## 2 KINDS OF CLASSIFIER :

1) discrete ⟶ yes or no (perceptron, SVM, DT)

2) probabilistic ⟶ P(yes/x) or P(no/x) (logistic regression)

↑

ROC curve only for this

(change threshold value θ to draw ROC curve)

$$\hat{y}_i = \begin{cases} 1 & P(y_i = 1|X) \geq \theta \\ 0 & \text{otherwise} \end{cases}$$

arrange in increasing order of X-axis to get smooth curve.

ALGORITHM

input $D = \{x_i, Y_i\}_{i=1}^{n}$

output : ROC curve

1. for each threshold $\theta$ , $0 \le \theta \le 1$

   a. for each $(x_i, Y_i) \in D$

      i. $\hat{Y}_i = \begin{cases} 1 & \text{if } P(Y_i = 1 \mid x) > \theta \\ 0 & \text{otherwise} \end{cases}$

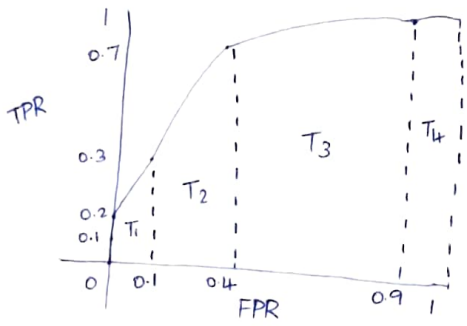
   b. calculate $TPR_\theta = \frac{TP}{P}$ , $FPR_\theta = \frac{FP}{N}$

2. arrange (TPR, FPR) in increasing order

* Plot the ROC curve with actual class labels and probability scores for classifiers.

| Y | $\hat{Y}$ | $\hat{Y}_{0.2}$ | $\hat{Y}_{0.4}$ | $\hat{Y}_{0.6}$ | $\hat{Y}_{0.8}$ | $\hat{Y}_{0.9}$ |
|---|---|---|---|---|---|---|
| P | 0.9 | P | P | P | P | P |
| P | 0.8 | P | P | P | P | N |
| N | 0.7 | P | P | P | N | N |
| P | 0.6 | P | P | P | N | N |
| P | 0.55 | P | P | N | N | N |
| P | 0.54 | P | P | N | N | N |
| N | 0.53 | P | P | N | N | N |
| N | 0.52 | P | P | N | N | N |
| P | 0.51 | P | P | N | N | N |
| N | 0.505 | P | P | N | N | N |
| P | 0.4 | P | P | N | N | N |
| N | 0.39 | P | N | N | N | N |
| P | 0.38 | P | N | N | N | N |
| N | 0.37 | P | N | N | N | N |
| N | 0.36 | P | N | N | N | N |
| N | 0.35 | P | N | N | N | N |
| P | 0.34 | P | N | N | N | N |
| N | 0.33 | P | N | N | N | N |
| P | 0.3 | P | N | N | N | N |
| N | 0.1 | N | N | N | N | N |
| TPR = $\frac{TP}{P}$ | | $\frac{10}{10} = 1$ | $\frac{7}{10} = 0.7$ | $\frac{3}{10} = 0.3$ | $\frac{2}{10} = 0.2$ | $\frac{1}{10} = 0.1$ |
| FPR = $\frac{FP}{N}$ | | $\frac{9}{10} = 0.9$ | $\frac{4}{10} = 0.4$ | $\frac{1}{10} = 0.1$ | 0 | 0 |

| | | | | | |
|---|---|---|---|---|---|
| FPR | 0 | 0 | 0.1 | 0.4 | 0.9 |
| TPR | 0.1 | 0.2 | 0.3 | 0.7 | 1 |




AUC = area of $(T_1 + T_2 + T_3 + T_4)$

area of $T_1 = 1/2 \times 0.1 \times 0.5 = 0.025$

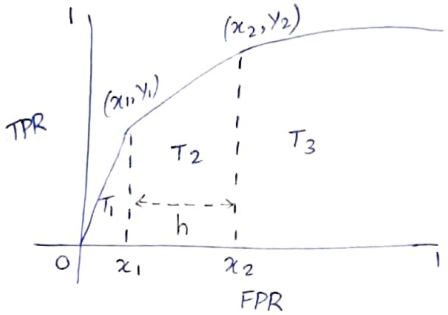
area of $T_2 = 1/2 \times 0.3 \times 1.0 = 0.15$

area of $T_3 = 1/2 \times 0.5 \times 1.7 = 0.425$

area of $T_4 = 1/2 \times 0.1 \times 2 = 0.1$

**AUC = 0.7**

## AREA UNDER THE ROC




**AUC of ideal model is 1**

area of trapezium = $1/2 \times$ height $\times$ (sum of parallel sides)

$= 1/2 \times (x_2 - x_1) \times (y_1 + y_2)$

$= 1/2 \times (FPR_2 - FPR_1) \times (TPR_1 + TPR_2)$

| DAY | OUTLOOK | TEMPERATURE | HUMIDITY | WIND | PLAY TENNIS |
|---|---|---|---|---|---|
| 1 | sunny | 85/hot | 85/H | weak | no |
| 2 | sunny | 80/hot | 90/H | strong | no |
| 3 | overcast | 83/hot | 78/H | weak | yes |
| 4 | rain | 70/mild | 96/H | weak | yes |
| 5 | rain | 68/cool | 80/N | weak | yes |
| 6 | rain | 65/cool | 70/N | strong | no |
| 7 | overcast | 64/cool | 65/N | strong | yes |
| 8 | sunny | 72/mild | 95/H | weak | no |
| 9 | sunny | 69/cool | 70/N | weak | yes |
| 10 | rain | 75/mild | 80/N | weak | yes |
| 11 | sunny | 75/mild | 70/N | strong | yes |

H - high N - normal

| | | | | | |
|---|---|---|---|---|---|
| 12 | overcast | 12/mild | 90/H | strong | yes |
| 13 | overcast | 21/hot | 75/N | weak | yes |
| 14 | rain | 11/mild | 80/H | strong | no |

NAIVE BAYES CLASSIFIER (probabilistic classifier)

Given

$$\mathcal{D} = \{x_i, y_i\}_{i=1}^{n}$$

$x_i \in$ discrete / continuous

$y_i \in \{C_1, C_2\}$

$f : x_i \longrightarrow y_i$

objective

$\hat{f} : \mathbb{R}^d \rightarrow \{C_1, C_2\}$

hypothesis

maximum aposterior (MAP) probability

$$\hat{y} = \underset{C_k}{\text{argmax}} \{P(C_k|X)\} \qquad C_k \in \{C_1, C_2\} \longrightarrow \text{map theorem}$$

$$\hat{y} = \begin{cases} C_1 & \text{if } P(C_1|X) \geq P(C_2|X) \\ C_2 & \text{otherwise} \end{cases}$$

assumption $P(F_1|Y, F_2, F_3 \ldots F_d) = P(F_1|Y)$

all features are conditionally independant given class

* From play tennis data set,

prior probability : without observing input features and based on class label if we calculate probability then it is called prior probability.

$P(yes) = 9/14 \qquad P(no) = 5/14$

posterior probability : after observing input features we obtain probability

$P(yes|x) \rightarrow$ conditional probability

It is a generative model

prior $P(yes)$ $P(no)$

posterior $P(yes|X)$ $P(no|X)$

joint $P(X, Y)$

marginal / total (sum of all joint probabilities) denominator of Bayes' theorem ($P(Y)$)

conditional $P(X|Y) = \frac{P(X \wedge Y)}{P(Y)}$ (or) $P(X|Y) = \frac{P(X,Y)}{P(Y)} = \frac{P(Y|X)P(X)}{P(Y)}$ (or) $\frac{\#_1}{\#_1}$ Bayes' theorem

eg: $P(O = sunny, PT = yes) = \frac{2}{14}$

$P(O = overcast, PT = yes) = \frac{4}{14}$

$P(O = rain, PT = yes) = \frac{3}{14}$

$P(O = sunny, PT = no) = 3/14$

$P(O = overcast, PT = no) = 0$

$P(O = rain, PT = no) = 2/14$

H/W Construct joint probability table and conditional probability for all features in dataset.

JPT:

| outlook | yes | no |
|---|---|---|
| sunny | 2/14 | 3/14 |
| overcast | 4/14 | 0 |
| rain | 3/14 | 2/14 |

| temperature | yes | no |
|---|---|---|
| hot | 2/14 | 2/14 |
| mild | 4/14 | 2/14 |
| cool | 3/14 | 1/14 |

| humidity | yes | no |
|---|---|---|
| high | 3/14 | 4/14 |
| normal | 6/14 | 1/14 |

| wind | yes | no |
|---|---|---|
| weak | 6/14 | 2/14 |
| strong | 3/14 | 3/14 |

CPT:

| outlook | yes | no |
|---|---|---|
| S | 2/9 | 3/5 |
| O | 4/9 | 0 |
| R | 3/9 | 2/5 |

| temperature | yes | no |
|---|---|---|
| H | 2/9 | 2/5 |
| M | 4/9 | 2/5 |
| C | 3/9 | 1/5 |

| | | yes | no |
|---|---|---|---|
| humidity | H | 3/9 | 4/5 |
| | N | 6/9 | 1/5 |

| | | | |
|---|---|---|---|
| wind | S | 3/9 | 3/5 |
| | W | 6/9 | 2/5 |

joint probability of 2 RV:

$$P(A,B) = P(A|B)P(B) \text{ (or) } P(B|A)P(A)$$

joint probability of 3 RV

$$P(A,B,C) = P(A|B,C)P(B,C)$$
$$= P(A|B,C)P(B|C)P(C)$$

joint probability of n RV: (by chain rule)

$$P(A_1, A_2 \ldots A_n) = P(A_1|A_2, A_3 \ldots A_n)P(A_2 A_3 \ldots A_n)$$
$$= P(A_1|A_2, A_3 \ldots A_n)P(A_2|A_3, A_4 \ldots A_n)P(A_3 A_4 \ldots A_n)$$
$$= P(A_1|A_2^n)P(A_2|A_3^n) \ldots P(A_{n-2}|A_{n-1}^n)P(A_{n-1}|A_n)P(A_n)$$
$$= \left[\prod_{j=1}^{n-1} P(A_j|A_{j+1}^n)\right] P(A_n)$$

NAIVE BAYES CLASSIFIER:

Given X, predict $\hat{Y}$

posterior probability

$$P(\hat{Y}=1|X) = \frac{P(X,\hat{Y}=1)}{P(X)} = \frac{P(X,\hat{Y}=1)}{P(X,\hat{Y}=0)+P(X,\hat{Y}=1)}$$

$$P(\hat{Y}=0|X) = \frac{P(X,\hat{Y}=0)}{P(X,\hat{Y}=0)+P(X,\hat{Y}=1)}$$

$$\hat{Y} = \begin{cases} 1 & P(\hat{Y}=1|X) \geq P(\hat{Y}=0|X) \\ 0 & \text{otherwise} \end{cases}$$

$$\hat{Y} = \underset{C_k}{\text{argmax}}\ P(C_k|X) \quad \text{where } C_k \in \{0,1\}$$

It is called Naive Bayes classifier, because of the naive conditional independance.

$$P(X_1|X_2, Y) = P(X_1|Y)$$

$$P(X_1|X_2, X_3, Y) = P(X_1|Y)$$

numerator of Bayes theorem:

$$P(x|y)\,P(y) = P(x, y)$$

$$= P(x_1, x_2, x_3, \ldots x_d, y)$$

$$= P(x_1 | x_2^d, y)\; P(x_2 | x_3^d, y) \ldots P(x_{d-1} | x_d, y)\, P(x_d | y) P(y)$$

By conditional independance,

$$= P(x_1|y) P(x_2|y) \ldots P(x_{d-1}|y)\, P(x_d|y) P(y)$$

$$= \left[\prod_{j=1}^{d} P(x_j|y)\right] P(y)$$

denominator of Bayes theorem:

$$P(x) = P(x_1, x_2, \ldots x_d)$$

$$= P(x, y=0) + P(x, y=1)$$

$$= P(x_1, x_2 \ldots x_d, y=0) + P(x_1, x_2 \ldots x_d, y=1)$$

$$= \prod_{j=1}^{d} P(x_j|y=0)\; P(y=0) + \prod_{j=1}^{d} P(x_j|y=1)\; P(y=1) \quad \text{(from numerator)}$$

$$P(y=C_1|x) = \frac{P(x, y=C_1)}{P(x, y=C_1) + P(x, y=C_2)}$$

$$= \frac{P(x_1|y=C_1) P(x_2|y=C_1) \ldots P(x_d|y=C_1) P(C_1)}{\prod_{j=1}^{d} P(x_j|y=C_1) P(y=C_1) + \prod_{j=1}^{d} P(x_j|y=C_2) P(y=C_2)}$$

$$P(y=C_2|x) = \frac{P(x, y=C_2)}{P(x, y=C_1) + P(x, y=C_2)}$$

$$= \frac{P(x_1|y=C_2) P(x_2|y=C_2) \ldots P(x_d|y=C_2) P(C_2)}{\prod_{j=1}^{d} P(x_j|y=C_1) P(y=C_1) + \prod_{j=1}^{d} P(x_j|y=C_2) P(y=C_2)}$$

model is conditional & prior probability. It is generative model because it constructs model for each class. It also finds joint probability. It tries to learn data space.

## TEST DATASET:

outlook = rainy  temperature = cool  humidity = normal

windy = strong

$P(PT = yes \mid <rainy, cool, normal, strong>)$

$$= \frac{P(OL = rainy \mid yes)\, P(T = cool \mid yes)\, P(H = normal \mid yes)\, P(W = strong \mid yes)\, P(yes)}{P(rainy \mid yes)\, P(cool \mid yes)\, P(normal \mid yes)\, P(strong \mid yes)\, P(yes) + P(rainy \mid no)\, P(cool \mid no)\, P(normal \mid no)\, P(strong \mid no)\, P(no)}$$

$$= \frac{3/9 \times 3/9 \times 6/9 \times 3/9 \times 9/14}{(3/9 \times 3/9 \times 6/9 \times 3/9 \times 9/14) + (2/5 \times 1/5 \times 1/5 \times 3/5 \times 5/14)}$$

$$= \frac{0.01587}{0.01587 + 0.00343} \simeq 0.822279793$$ (Play tennis = yes)

## Buys computer dataset: (H/W)

## Sunburn dataset:

| hair | height | weight | lotion | result |
|---|---|---|---|---|
| blonde | average | light | no | sunburned |
| blonde | tall | average | yes | none |
| brown | short | average | yes | none |
| blonde | short | average | no | sunburned |
| red | average | heavy | no | sunburned |
| brown | tall | heavy | no | none |
| brown | average | heavy | no | none |
| blonde | short | light | yes | none |

CPT:

| hair | result | |
|---|---|---|
| | sunburned | none |
| blonde | 2/3 | 2/5 |
| brown | 0/3 | 3/5 |
| red | 1/3 | 0/5 |

| height | result | |
|---|---|---|
| | sunburned | none |
| short | 1/3 | 2/5 |
| average | 2/3 | 1/5 |
| tall | 0/3 | 2/5 |

| | | result | |
|---|---|---|---|
| weight | light | 1/3 | 1/5 |
| | average | 1/3 | 2/5 |
| | heavy | 1/3 | 2/5 |

| | | result | |
|---|---|---|---|
| lotion | yes | 0/3 | 3/5 |
| | no | 3/3 | 2/5 |

Predict test data for patient with brown hair, tall, heavy weight & uses lotion.

P(yes | <brown, tall, heavy, yes>)

$$= \frac{P(\text{brown}|\text{yes})\,P(\text{tall}|\text{yes})\,P(\text{heavy}|\text{yes})\,P(\text{yes}|\text{yes})\,P(\text{yes})}{P(\text{brown}|\text{yes})\,P(\text{tall}|\text{yes})\,P(\text{heavy}|\text{yes})\,P(\text{yes}|\text{yes})\,P(\text{yes}) + P(\text{brown}|\text{no})\,P(\text{tall}|\text{no})\,P(\text{heavy}|\text{no})\,P(\text{yes}|\text{no})\,P(\text{no})}$$

= 0 (since insufficient dataset)

※ disadvantage of Naive Bayes

2 CHALLENGES:

* zero frequency
* continuous features

Zero frequency problem:

$F_1, F_2 \ldots F_d$ are features

$F_j = \{v_1, v_2 \ldots v_m\}$

$$P(F_j = v_m | Y_k) = \frac{n_{jmk}}{n_k}$$

where

$n_{jmk}$ = # times the feature $F_j$ takes value $v_m$ in class $Y_k$

$n_k$ = # datapoints in class $C_k$

If $n_{jmk} = 0$, $P(F_j = v_m | Y_k) = \frac{n_{jmk} + 1/m}{n_k + 1}$

eg: outlook

| | yes | | no | |
|---|---|---|---|---|
| S | 2/9 | | 3/5 | $\frac{3+1/3}{6}$ |
| O | 4/9 | | 0 | $\frac{0+1/3}{6}$ |
| R | 3/9 | | 2/5 | $1/3$ $2/5+1$ |

Smoothening technique

1) laplacian (added 1/m)
2) added one

By laplacian technique,

CPT:

| hair | | result | |
|---|---|---|---|
| | blonde | 7/12 | 7/18 |
| | brown | 1/12 | 10/18 |
| | red | 4/12 | 1/18 |

| height | | result | |
|---|---|---|---|
| | short | 4/12 | 2/5 |
| | average | 7/12 | 1/5 |
| | tall | 1/12 | 2/5 |

| weight | | | |
|---|---|---|---|
| | light | 1/3 | 1/5 |
| | average | 1/3 | 2/5 |
| | heavy | 1/3 | 2/5 |

| lotion | | | |
|---|---|---|---|
| | yes | 1/8 | 3/5 |
| | no | 7/8 | 2/5 |

$P(yes \mid <brown, tall, heavy, yes>)$

$$= \frac{1/12 \times 1/12 \times 1/3 \times 1/8 \times 3/8}{(1/12 \times 1/12 \times 1/3 \times 1/8 \times 3/8) + (10/18 \times 2/5 \times 2/5 \times 3/5 \times 5/8)}$$

$= 0.0032476$ (result = none)

$$\frac{0.0001085}{0.0001085 + 0.0333}$$

Buy computer dataset :

| age | income | student | credit rating | buy |
|---|---|---|---|---|
| youth | high | no | fair | no |
| youth | high | no | excellent | no |
| middle_age | high | no | fair | yes |
| senior | medium | no | fair | yes |
| senior | low | yes | fair | yes |
| senior | low | yes | excellent | no |
| middle_age | low | yes | excellent | yes |
| youth | medium | no | fair | no |
| youth | low | yes | fair | yes |
| senior | medium | yes | fair | yes |
| youth | medium | yes | excellent | yes |
| middle_age | medium | no | excellent | yes |
| middle_age | high | yes | fair | yes |

CPT:

| | | yes | no |
|---|---|---|---|
| age | youth | 2/9 | 3/4 |
| | middle-age | 4/9 | 0/4 |
| | senior | 3/9 | 1/4 |

By add one,

| | |
|---|---|
| 2/9 | 10/15 |
| 4/9 | 1/15 |
| 3/9 | 4/15 |

| | | | |
|---|---|---|---|
| income | high | 2/9 | 2/4 |
| | low | 3/9 | 1/4 |
| | medium | 4/9 | 1/4 |

| | | | |
|---|---|---|---|
| student | yes | 6/9 | 1/4 |
| | no | 3/9 | 3/4 |

| | | | |
|---|---|---|---|
| credit rating | fair | 6/9 | 2/4 |
| | excellent | 3/9 | 2/4 |

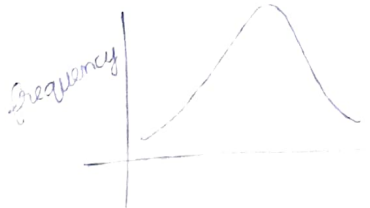
$P(no \mid <medium, senior, no, excellent>)$

$$= \frac{1/4 \times 1/4 \times 3/4 \times 2/4 \times 4/13}{(1/4 \times 1/4 \times 3/4 \times 2/4 \times 4/13) + (3/9 \times 4/9 \times 3/9 \times 3/9 \times 9/13)} = \frac{0.00721}{0.00721 + 0.011396}$$

$= 0.3875$ ( buy computer = yes)

continuous features:

* convert to discrete values / discretisation)
* assume feature follows gaussian / normal dist




using pdf construct CPT

$$\text{Pdf} = \frac{1}{\sqrt{2\pi}\,\sigma} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

known : X

unknown : $\mu, \sigma$

| day | outlook | temperature | humidity | wind | play |
|---|---|---|---|---|---|
| 1 | sunny | 85 | 85 | weak | no |
| 2 | sunny | 80 | 90 | strong | no |
| 3 | overcast | 83 | 78 | weak | yes |
| 4 | rain | 70 | 96 | weak | yes |
| 5 | rain | 68 | 80 | weak | yes |
| 6 | rain | 65 | 70 | strong | no |
| 7 | overcast | 64 | 65 | strong | yes |
| 8 | sunny | 72 | 95 | weak | no |
| 9 | sunny | 69 | 70 | weak | yes |
| 10 | rain | 75 | 80 | weak | yes |
| 11 | sunny | 75 | 70 | strong | yes |
| 12 | overcast | 72 | 90 | strong | yes |
| 13 | overcast | 81 | 75 | weak | yes |
| 14 | rain | 71 | 80 | strong | no |

test data : sunny , 40, 70, strong

temperature | play tennis = yes

$\bar{x} = 73$ $\sigma = \sqrt{\frac{\sum(x_i - \bar{x})^2}{n-1}} = 6.164414$

temperature | play tennis = no

$\bar{x} = 74.6$ $\sigma = 7.8930349$

humidity | play tennis = yes

$\bar{x} = 78.222$ $\sigma = 9.88407811$

humidity | play tennis = no

$\bar{x} = 84$ $\sigma = 9.61769203$

CPT:

| | yes | no |
|---|---|---|
| temperature | $3.87 \times 10^{-8}$ | $2.27 \times 10^{-6}$ |
| humidity | 0.028594 | 0.014271 |

yes | < sunny, 40, 70, strong >)

P(sunny|yes) P(40|yes) P(70|yes) P(strong|yes) P(yes)

(sunny|yes) P(40|yes) P(70|yes) P(strong|yes) P(yes) +

(sunny|no) P(40|no) P(70|no) P(strong|no) P(no)

$$\frac{2/9 \times (3.87\times10^{-8}) \times (0.028599) \times 3/9 \times 9/14}{\left[(2/9)\times(3.87\times10^{-8})\times(0.028599)\times 3/9 \times 9/14\right] + \left[(3/5)\times(3.37\times10^{-6})\times(0.014371)\times 3/5 \times 5/14\right]}$$

$$= \frac{5.27\times10^{-11}}{(5.27\times10^{-11}) + (6.2267\times10^{-9})}$$

= 0.00839 (play tennis = no)

## MULTIVARIATE GAUSSIAN:

Let $x_1, x_2 \ldots x_k$ be k features

$x_1 \sim N(\mu_1, \sigma_1)$

$x_2 \sim N(\mu_2, \sigma_2)$

$\vdots$

$x_k \sim N(\mu_k, \sigma_k)$

$$ax_1 + bx_2 \sim N(\mu, \Sigma)$$

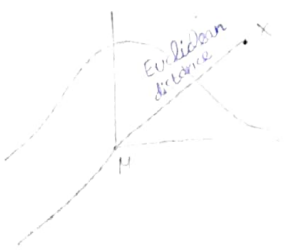
where $\mu$ is mean vector of len k

$\Sigma$ is covar matrix of len k×k

$$P(x) = \frac{1}{(2\pi)^{k/2}|\Sigma|^{1/2}} e^{-1/2(x-\mu)^T \Sigma^{-1}(x-\mu)}$$

z score gaussian (or) Mahalanobis distance

$$\Sigma = \begin{bmatrix} \sigma_1^2 & \sigma_{12} & \cdots & \sigma_{1k} \\ \sigma_{21} & \sigma_2^2 & \cdots & \sigma_{2k} \\ \vdots & & & \vdots \\ \sigma_{k1} & \sigma_{k2} & \cdots & \sigma_{kk} \end{bmatrix} \qquad |\Sigma| = \sigma_1^2 \sigma_2^2 \ldots \sigma_k^2$$




distance between x and μ.

$\because (x-\mu)^2 = (x-\mu)^T(x-\mu)$ w.r.t $\Sigma$ (covar)

If $\Sigma$ is a unit matrix, then

Mahalanobis dist = Euclidean dist

$x_{k\times1}$ $M_{k\times1}$ $\Sigma_{k\times k}$

Consider the class means $\mu$ and $\Sigma$ for classes $C_1$ and $C_2$

$\mu_1 = 1, 3$ $\quad$ $\mu_2 = 5, 5$

$\Sigma_1 = \begin{matrix} 5 & 3 \\ 3 & 2 \end{matrix}$ $\quad$ $\Sigma_2 = \begin{matrix} 2 & 0 \\ 0 & 1 \end{matrix}$

Classify 3,4 using NB, full NB

$P(\text{class } C_1) = 1/4$ $\quad$ $P(\text{class } C_2) = 3/4$

$$P(\text{class } C_1 | <3,4>) = \frac{P(<3,4> | C_1)\, P(C_1)}{P(<3,4>)}$$

$$P(<3,4> | C_1) = \frac{1}{(2\pi)^{2/2} (10)^{1/2}} e^{-1/2 \begin{bmatrix} 2 & 1 \end{bmatrix} \begin{bmatrix} \frac{2}{10} & 0 \\ 0 & 5/10 \end{bmatrix} \begin{bmatrix} 2 \\ 1 \end{bmatrix}}$$

$$|\Sigma_1| = \begin{vmatrix} 5 & 0 \\ 0 & 2 \end{vmatrix} = 10$$ (By conditional independence assumption)

$$P(<3,4> | C_1) = 0.02628 \times 1/4 = 0.0065718$$

$$\begin{bmatrix} 4/10 & 5/10 \end{bmatrix} \begin{bmatrix} 2 \\ 1 \end{bmatrix}$$

$$P(<3,4> | C_2) = \frac{P(<3,4> | C_2)\, P(C_2)}{P(<3,4>)} = \frac{13}{10}$$

$$P(<3,4> | C_2) = \frac{1}{(2\pi)^{2/2} (2)^{1/2}} e^{-1/2 \begin{bmatrix} -2 & -1 \end{bmatrix} \begin{bmatrix} 1/2 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} -2 \\ -1 \end{bmatrix}}$$

$$P(<3,4> | C_2) = 0.02512$$

$$P(<3,4> | C_2) = 0.02512 \times 3/4 = 0.018843$$

$$\begin{bmatrix} -1 & -1 \end{bmatrix} \begin{bmatrix} -2 \\ -1 \end{bmatrix} = 3$$

$\therefore$ prediction : $C_2$

## * PRIVACY PRESERVING ML ALGORITHM:

model 1 - model for n datapoints

model 2 - model for n-1 datapoints

(build for n datapoints, remove 1 datapoint randomly)

If model 1 and model 2 are same then it is called privacy preserving ML.

eg: differentially private kNN, decision trees

* Add gaussian noise or laplacian noise to X or Y. It is called perturbation

↓

It is done to the data stored in cloud in order to attain privacy.

* security: suppress the importance of wrong data also due to

eg: robust KNN

added by attacker / hacker to suppress efficiency of algorithm

## EXPLAINABLE ML ALGORITHM / EXPLAINABILITY:

The model should predict the outcome also explain on why it is predicted that way.

## K-NEAREST NEIGHBOURS (KNN)

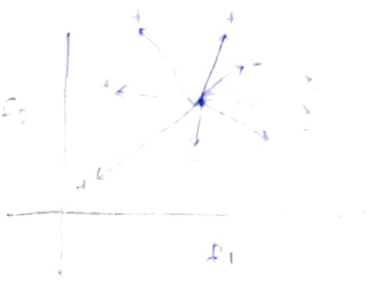
- classification
- regression

* time consuming as model is constructed in testing phase
* no training phase (memory based model, training set in memory)
* lazy learners (training set req at testing phase) model constructed for each data point. diff model for diff data point (KNN) since it is diff, model is constructed at the time of testing.
* Other algorithms are eager learners - model constructed in training phase (no training set required, ~~no~~ only model required at testing)
* instance based

At the time of testing, find the distance between the test data and other training set data points.

$X = [a, b]$ (test data)

$X_i = [a_i, b_i]$ (data points)



$X = \{a, b, c, \ldots\}$

$X_1 = \{a_1, b_1, c_1, \ldots\}$

$dis(X, X_1) = \sqrt{\sum_{j=1}^{d} (x_j - x_{1j})^2}$

↓

can use Euclidean dist / Mahattan dist / Minkowski dist / Chebyshev dist / Mahalanobis dist

k - neighbours, k - hyperparameter

Find optimal value of k by three way split

$k = 5 \quad C_+ = 2 \quad C_- = 3$

test data should be assigned to -ve class

avoid even values for k incase of tie

ASSUMPTION :

closest data points belong to same class (closeness)

Regression : Don't do voting. Y value of test data is the mean of y values of k nearest neighbours.

can be any measure

| $X_1$ = acid durability (seconds) | $X_2$ = strength ($kg/m^2$) | Y = classification |
|---|---|---|
| 7 | 7 | bad |
| 7 | 4 | bad |
| 3 | 4 | good |
| 1 | 4 | good |

Predict test data (2, 2) & (5, 6) using training set with $k = 3$ by applying KNN.

* (2,2)

$\sqrt{(x_2-x_1)^2 + (y_2-y_1)^2}$

| training data | distance | class |
|---|---|---|
| (7,7) | 7.07106781 | |
| (7,4) | 5.38516 | bad |
| (3,4) | 2.236 | good |
| (1,4) | 2.236 | good |

k neighbours, K=3

∴ (2,2) is predicted good

* (5,6)

| training data | distance | class |
|---|---|---|
| (7,7) | 2.236 | bad |
| (7,4) | 2.828 | bad |
| (3,4) | 2.828 | good |
| (1,4) | 4.4721 | |

∴ (5,6) is predicted bad

ALGORITHM - KNN: (pre-processing ※ & normalisation)

input

$\mathcal{D} = \{x_i, y_i\}_{i=1}^{n}$

output

1 or 0 (classification)

k, test data (x, y)

1. for each test data $(x, y) \in$ test

a. $C_0 = C_1 = 0$

b. for each train data $(x_i, y_i) \in \mathcal{D}$

i. $dist_i$ = distance $(x_i, x)$ — O(nd)

c. sort $dist_i$ — O(n log n)

d. $N = \{\langle x_i, y_i \rangle \mid dist_i \text{ is minimum}\}$ & $|N| = k$ — O(k)

e. $C_0 = \sum_{x_i, y_i \in N} 1(y_i == 0)$ — O(k)

$C_1 = \sum_{x_i, y_i \in N} 1(y_i == 1)$ — O(k)

e. $\hat{y} = \frac{\sum_{x_i, y_i \in N} y_i}{k}$ (regression) → mean

f. $\hat{Y} = \begin{cases} 1 & \text{if } C_1 > C_0 \\ 0 & \text{o/w} \end{cases}$ — O(1)

for one test data,

time complexity = max (nd, nlog n)

Normalisation technique:

z-score normalisation

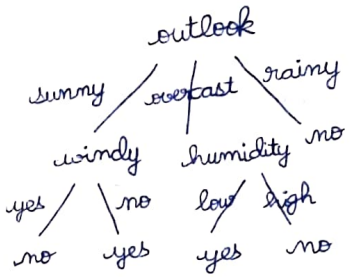
min-max normalisation

Model: K nearest neighbours / closed polygon covers all K neighbours)

DECISION TREES:

classification & regression

model - tree




* interior nodes - features
* branches - possible values
* leaf nodes - classes

pure / homogenous

If all datapoints in D belong to same class.

objective

given $D = \{x_i, y_i\}_{i=1}^{n}$ & impure

It aims to convert D into pure by splitting D based on features.

impurity measures

- entropy
- conditional entropy
- information gain (ID3) iterative dicotamizer
- gain ratio (C4.5)
- gaini index (CART) classification & regression tree

entropy $\boxed{0 \leq H(Y) \leq \log_2 m}$ (surprisal)

* impurity of data
* information content in a RV
* randomness
* # bits required to represent a RV (from information theory)

(information $\propto$ 1/probability)

| when Y | H(Y) entropy (no of bits to represent) |
|---|---|
| 2 | 1 |
| 4 | 2 |
| . | . |
| . | . |
| m | $\log_2 m$ |

equally likely

$H(Y) = \log_2 m = -\log_2(1/m) = -\log P$ (unbiased)

If Y is biased,

$$H(Y) = -\sum_{i=1}^{m} P_i \log P_i$$

$E(Y) = Y H(Y)$

mean

Calculate entropy of play tennis, buy computer & iris dataset.

* play tennis $= -\frac{9}{14}\log_2\frac{9}{14} - \frac{5}{14}\log_2\frac{5}{14} = 0.94$

* buy computer $= -\frac{8}{13}\log\frac{8}{13} - \frac{5}{13}\log\frac{5}{13} = 0.9611$

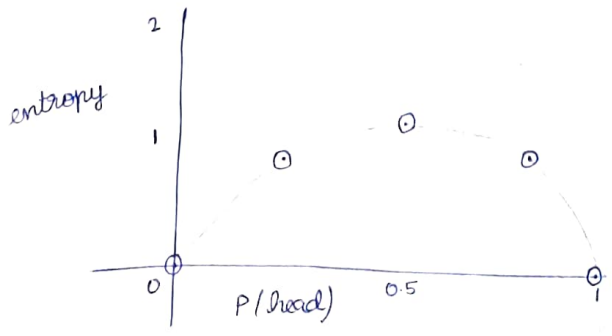
* iris dataset $= -\log 1/3 = 1.58$ (unbiased)

Note: $\log 0 = 0$

for pure dataset, entropy is 0.

H/W Plot the curve for entropy of Bernoulli RV by finding entropy of probability distribution

| P(head) | P(tail) | pmf $p^x(1-p)^{1-x}$ <br> entropy |
|---|---|---|
| 0 | 1 | $-0\log 0 - 1\log 1 = 0$ |
| 0.25 | 0.75 | $-0.25\log 0.25 - 0.75\log 0.75 = 0.8113$ |
| 0.50 | 0.50 | $-0.50\log 0.50 - 0.50\log 0.50 = 1$ |
| 0.75 | 0.25 | $-0.75\log 0.75 - 0.25\log 0.25 = 0.8113$ |
| 1 | 0 | $-1\log 0 - 0\log 1 = 0$ |




concave function

Find using KNN

| H/W | 1 | 2 | 3 | 4 | |
|---|---|---|---|---|---|
| CASES OF PRODUCT | 7 | 3 | | | 330 |
| DISTANCE (IN FT) | 560 | 220 | 340 | 80 | 150 |
| TIME (IN MINS) | 16.68 | 11.50 | 12.03 | 14.88 | 13.75 |

18.11 ↑ test data

(7, 330) training data

distance (1.32, 0.32)

(7, 560) 1.22

(3, 220) 0.442 ✓

(3, 340) 0.1961 ✓

(4, 80) 1.207

(6, 150) 0.636 ✓

check

$(x-\mu/\sigma)$ normalised

$\mu = 46$ $\sigma = 1.82$

$\mu = 270$ $\sigma = 188.41$

(1.32, 1.54)

(+0.88, +0.27)

(+0.88, 0.37)

(+0.33, +1.01)

(0.77, +0.64)

11.50

12.03

13.75

12.43 mean

MODE : SD , M+ , SHIFT 2

$\bar{x}$ - mean $\sigma x$ - std

dataset = { pure $H(Y)=0$ ; impure otherwise }

prob = high info = low # bits = 0

OBJECTIVE :

trying to minimise entropy (close to 0)

∵ entropy of pure dataset = 0

$H(Y|X)$ → conditional entropy

$H(Y|X = value)$ → specific conditional entropy (SCE)

$$H(Y|X=v) = -\sum_{i=1}^{m} P(Y_i|x=v) \log P(Y_i|x=v)$$

SCE : average number of bits req to represent Y when X takes some specific value.

* Entropy takes maximum value when it is equally distributed 1/m

* Entropy is minimum when it is pure dataset

eg: Calculate of SCE of play tennis given outlook = sunny, rainy and overcast.

SC entropy

**OUTLOOK:**

OL = sunny (5): yes 2, no 3 — $-\left(\frac{2}{5}\log\frac{2}{5} + \frac{3}{5}\log\frac{3}{5}\right) = 0.97$

OL = rainy: yes 3, no 2 — 0.97

OL = overcast: yes 4, no 0 — 0 (pure dataset)

**CONDITIONAL ENTROPY:** $0 \leq H(Y|X) \leq$ entropy of Y $H(Y)$

expectation of SCE

$$H(Y|X) = \sum_{v \in X} P(v)\, H(Y|X=v)$$

$$H(PT|OL) = P(S)\,H(PT|S) + P(R)\,H(PT|R) + P(OC)\,H(PT|OC)$$

$$= \frac{5}{14} \times 0.97 + \frac{5}{14} \times 0.97 + \frac{4}{14} \times 0$$

$$H(PT|OL) = 0.693$$

※ upper bound is always entropy of Y

SCE

**TEMPERATURE:**

T = hot (4): yes 2, no 2 — $-\left(\frac{2}{4}\log\frac{2}{4} + \frac{2}{4}\log\frac{2}{4}\right) = 1$

T = mild (6): yes 4, no 2 — $-\left(4/6\log 4/6 + 2/6\log 2/6\right) = 0.9183$

T = cold (4): yes 3, no 1 — $-\left(3/4\log 3/4 + 1/4\log 1/4\right)$ 0.81125

SCE: $H(PT|T) = P(hot)\,H(PT|hot) + P(mild)\,H(PT|mild) + P(cold)\,H(PT|cold)$

$$= \frac{4}{14} \times 1 + \frac{6}{14} \times 0.9183 + \frac{4}{14} \times 0.81125 = 0.91106$$

Humidity :

$H = \text{high}\ (7)$ — yes 3, no 4 $\quad -(+3/7 \log 3/7 + 4/7 \log 4/7) =$ 0.98526

$H = \text{normal}\ (7)$ — yes 6, no 1 $\quad -(6/7 \log 6/7 + 1/7 \log 1/7) =$ 0.5916714

CE: $\frac{7}{14}(0.9852) + \frac{7}{14}(0.591674) = 0.788451$

wind :

$W = \text{weak}\ (8)$ — yes 6, no 2 $\quad -(6/8 \log 6/8 + 2/8 \log 2/8) =$ 0.81127

$W = \text{strong}\ (6)$ — yes 3, no 3 $\quad -(3/6 \log 3/6 + 3/6 \log 3/6) =$ 1

CE: $\frac{8}{14}(0.81127) + \frac{6}{14}(1) = 0.8921$

NOTE :

* When $H(Y|X) = 0$, purity of dataset increases which means X highly influences Y. highly correlated.
* When $H(Y|X) = H(Y)$, X and Y are independant, X does not influence Y.

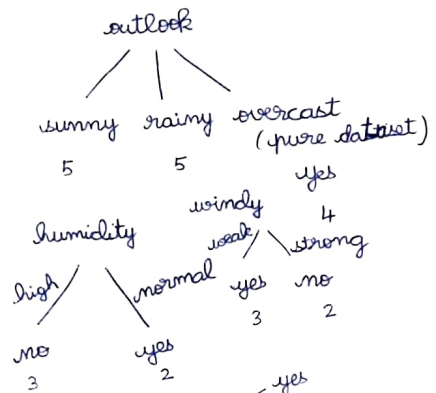
INFORMATION GAIN: (ID3 algorithm)

average # bits saved to represent Y using X

$$IG(X,Y) = H(Y) - H(Y|X)$$

$$\text{best feature} = \begin{cases} \text{argmax } IG(X,Y) \\ \text{or} \\ \text{argmin } H(Y|X) \end{cases} \quad X^*$$

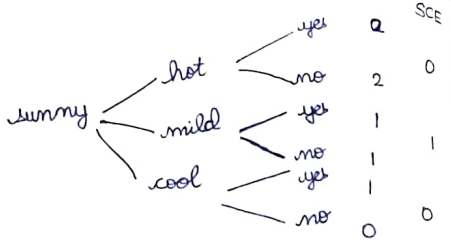
which feature reduces entropy the most will be root node



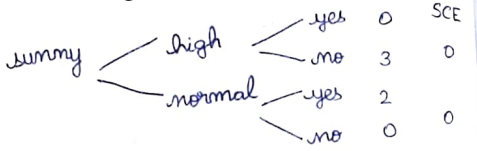

$H(PT|T) = 0.4$

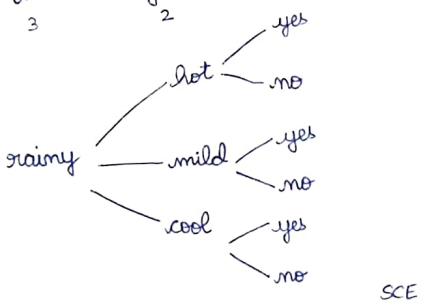
$H(PT|H) = 0$

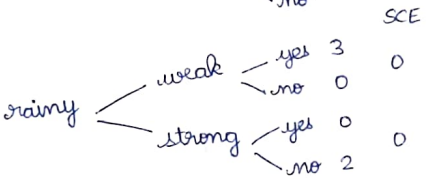
$H(PT|W) = 0.95$



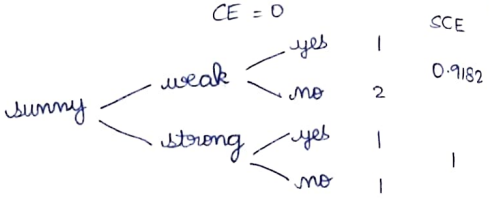

$CE = 1 \times 2/5 = 0.4$




$CE = 0$




$CE = (0.9182)\ 3/5 + 2/5\ (1)$

$= 0.9509$

CHARACTERISTICS: * non-parametric, discriminative

* Decision trees can be converted to sequence of if-then rules. Number of if-then is equal to number of leaf nodes

* There are 2 views: data view, feature view

  - data view → interior nodes : impure; leaf nodes : pure
  - feature view → features, class labels

* training error of decision trees is zero. overfitting
  low bias, high variance (SOLUTION: tree pruning)
  cutting leaf (edges)

* feature selection of pre-processing is not needed
  (decision tree automatically selects feature)

ML LAB: Construct decision tree for
Find height of tree and all performance
choose 5 random subset of features
Compare with original decision tree

NOTE:

* Inclusion of noise & outlier wil
due to overfitting

* To overcome overfitting we can
since it can be applied only to p

* Assume that we include SNo as a
tennis dataset. Find the inform
what would be the root node of
in the dataset?

In SNo, all have SCE as 0. ∴
SNo will be the root node (
Eventually $IG = H(Y) - H(Y|X)$ is
$IG = H(Y) = 0.94$

* IG gives high value which has
SNo doesn't influence PT, IG
* ID3 can be applied only for
↓
limitations.

GAIN RATIO: (C4.5)

$GR(X,Y) = \frac{IG(X,Y)}{H(X)}$ (norm

* can be applied for contin
* non influencial due to n

ML LAB : Construct decision tree for TIC TAC TOE dataset. Find height of tree and all performance measures. Then choose 5 random subset of features and construct DT. Compare with original decision tree. Find training err.

NOTE :

* Inclusion of noise & outlier will not give accurate DT due to overfitting

* To overcome overfitting we cannot use regularization since it can be applied only to parametric model

* Assume that we include SNo as a feature in play tennis dataset. Find the information gain of SNo and what would be the root node ~~of~~ if SNo is included in the dataset ?

In SNo, all have SCE as 0. ∴ CE of PT given SNo = 0

SNo will be the root node (∵ min CE)

Eventually $IG = H(Y) - H(Y|X)$ is maximum

$IG = H(Y) = 0.94$

* IG gives high value which has more splits. Eventhough SNo doesn't influence PT, IG choose SNo as best feature.

* ID3 can be applied only for discrete features

↓ limitations.

GAIN RATIO : (C4.5)

$$GR(X,Y) = \frac{IG(X,Y)}{H(x)}$$ (normalised information gain)

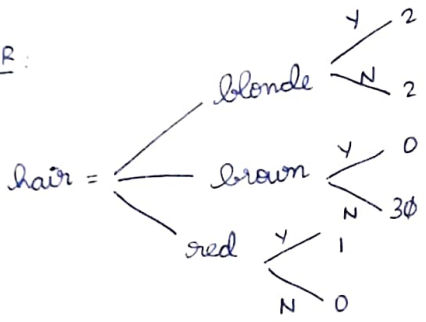
* can be applied for continuous features.

* non influencial due to more splits.

* Construct DT using C4.5 for sun burn dataset.

$H(Y) = 0.9544$

HAIR:




SCE

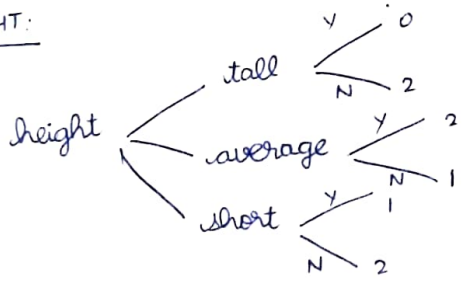
1

0

0

$CE = \frac{4}{8} \times 1 = \frac{1}{2}$

HEIGHT:




SCE

0

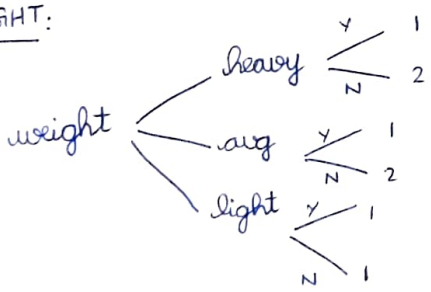
0.9182

0.9182

$CE = \left(\frac{3}{8} \times 0.9182\right) + \left(\frac{3}{8} \times 0.9182\right)$

$= 0.68865$

WEIGHT:




SCE

0.9182

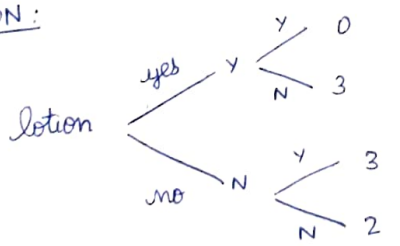
0.9182

1

$CE = \left(\frac{3}{8} \times 0.9182\right) + \left(\frac{3}{8} \times 0.9182\right) + \left(\frac{2}{8} \times 1\right)$

$= 0.93865$

LOTION:




SCE

0

0.97095

$CE = \left(\frac{5}{8} \times 0.97095\right)$

$= 0.6068$

| gain ratio | hair | height | weight | lotion |
|---|---|---|---|---|
| $\frac{IG(X,Y)}{H(X)}$ | $\frac{0.4544}{1.4056}$ | $\frac{0.26575}{1.56127}$ | $\frac{0.01575}{1.56127}$ | $\frac{0.3476}{0.9544}$ |
| | 0.3232 | 0.17021 | 0.010087 | 0.36420 |

✓ root node

## GINI INDEX : (CART algo)

easy to compute - uses log

$$GI = 1 - \sum_{i=1}^{m} p_i^2$$

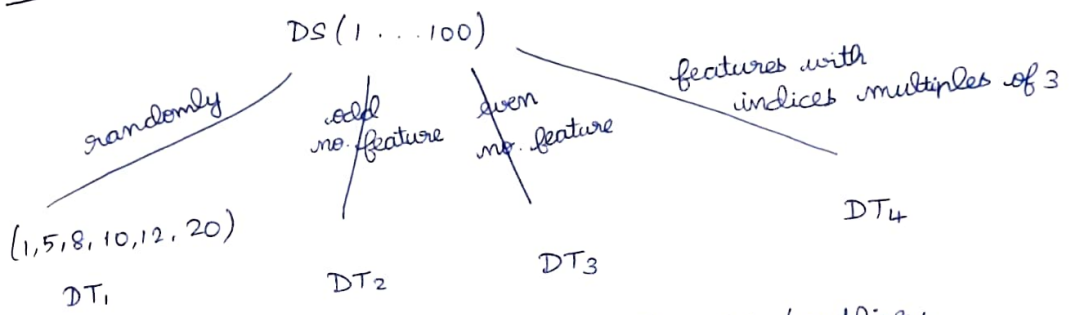
for pure dataset $GI = 0$

$$X^* = \underset{x}{\text{argmin}} \; GI(x, Y)$$

## PRUNING :

cut branches of leaf nodes by applying chi-square test. If it has high confidence values don't cut. (goodness of fit)

## RANDOM FOREST:



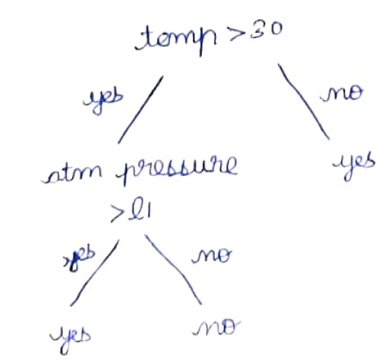

predict using all trees to avoid noise / outliers

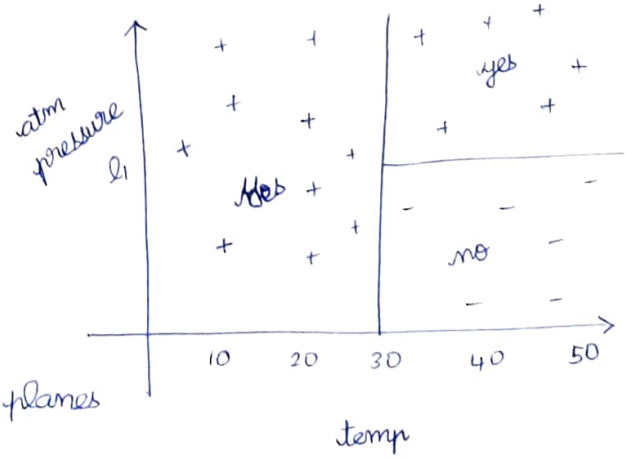
- collection of decision trees
- randomly choose subset of features / no. of datapoints

K times { $D_1$ = random() // $F_1, F_2, \ldots F_d$
construct ID3 ($D_1$) }

K = # decision trees (hyperparameter)

## GEOMETRIC REPRESENTATION OF DECISION TREES:




combination of parallel axis planes

## SUPPORT VECTOR MACHINE : (SVM) (parametric discriminative linear)

Given

$$D = \{x_i, y_i\}_{i=1}^{n} \qquad x_i \in \mathbb{R}^d \qquad y_i \in \{+1, -1\}$$

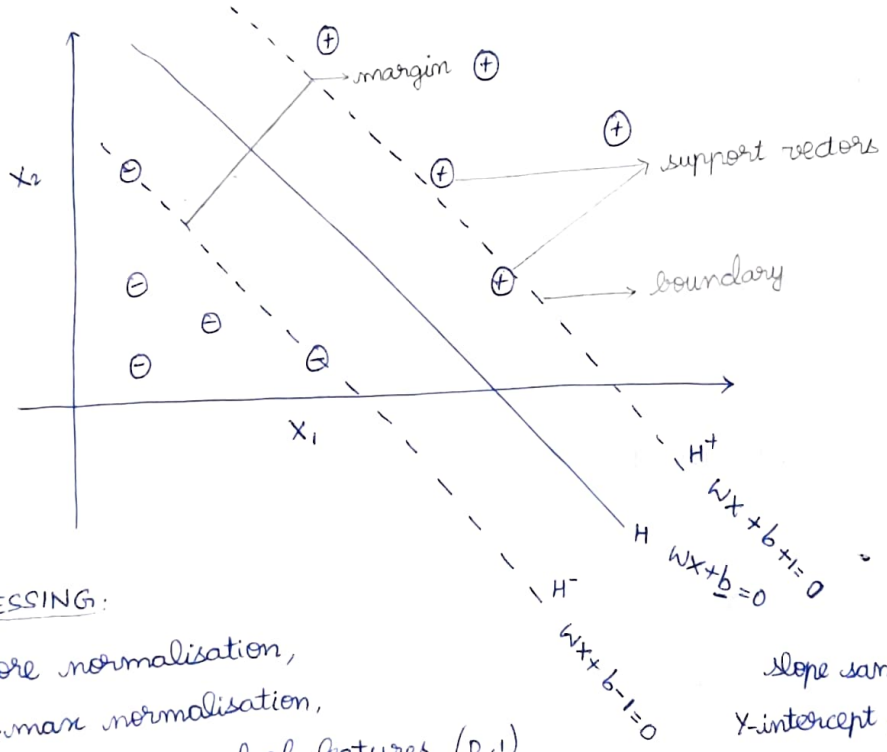
$$f : x_i \rightarrow y_i$$

Objective

$$\hat{f} : \mathbb{R}^d \longrightarrow \{+1, -1\}$$

* find an optimal hyperplane equi-distance from +ve & -ve margin
* max distance betw HP & boundary to avoid bias

VC dimension : (Vapnik, Chervonenki) max no. of datapoints req, to compute measures




Slope same
Y-intercept diff

PRE-PROCESSING :

* z score normalisation,
* min-max normalisation,
* divide by max val of features (0,1)

$$\hat{y} = \begin{cases} +1 & wx + b \geq 0 \\ -1 & wx + b < 0 \end{cases}$$

$$\boxed{\hat{y} = \text{sign}\,(wx + b)}$$

MODEL :

| dimension | model |
|---|---|
| 1 | point |
| 2 | line |
| 3 | plane |
| >3 | hyperplane |

max margin hyperplane

Optimal hyperplane is H with max margin

max = distance btw boundaries

Find W & b which maximizes margin

M = distance between $H_+$ & $H_-$

$H_+$ and $H_-$ are parallel

distance between parallel lines

$$wx + (b-1) = 0$$

$$wx + (b+1) = 0$$

$$M = \frac{|(b-1) - (b+1)|}{\|w\|} = \frac{2}{\|w\|}$$

for simplicity unit distance

Max $\frac{2}{\|w\|}$ S.t.c $wx_i + b \geq 1$ for $y_i \in +1$ n constraints

$wx_i + b \leq -1$ for $y_i \in -1$ for each datapoint

S.t.c

$$y_i (w x_i + b) \geq 1$$

$\Downarrow$

max margin = min $\|w\|$

$\Downarrow$

Min $1/2 \|w\|$

S.t.c

$y_i (wx_i + b) \geq 1$

min $1/2 w^T w$

S.t.c

$y_i (wx_i + b) \geq 1$

constrained convex optimisation problem

(Solve by lagrange multipliers & KKT)

Karush, Kuhn, Tucker

$\Downarrow$

lagrange multiplier

f: Min $1/2 \underbrace{w^T w}_{w^2} - \sum_{i=1}^{n} \alpha_i (y_i (wx_i + b) - 1)$ ① $\alpha_i \geq 0$ PRIMAL

② $\frac{\partial f}{\partial w} = w - \sum_{i=1}^{n} \alpha_i y_i x_i = 0$

$$w = \sum_{i=1}^{n} \alpha_i y_i x_i$$

③ $\frac{\partial f}{\partial b} = - \sum \alpha_i y_i = 0$

$$\sum \alpha_i y_i = 0$$

④ $\alpha_i (y_i (wx_i + b) - 1) \geq 0$

KKT conditions : ①, ②, ③, ④

Min

$$\frac{1}{2} w^T w - \sum_{i=1}^{n} \alpha_i y_i w x_i - \sum_{i=1}^{n} \alpha_i y_i b + \sum_{i=1}^{n} \alpha_i$$

by (2) by (3)

$$\frac{1}{2} w^T w - w^T w - 0 + \sum \alpha_i$$

$$-\frac{1}{2} w^T w + \sum \alpha_i$$

**DUAL**

$$\underset{w,b}{Min} = \underset{\alpha}{Max} \; -\frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i y_i x_i \alpha_j y_j x_j + \sum \alpha_i \quad \text{(function of } \alpha)$$

$\alpha_i, \alpha_j, y_i, y_j$ - scalars

$x_i, x_j$ - vectors

s.t.c

$\alpha_i \geq 0$

$\sum \alpha_i y_i = 0$

$$-\frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j x_i x_j$$

Note: $\alpha_i \neq 0$ then data point is said to be support vector

$\alpha_i = 0$ then data point is not support vector

$$SV = \{(x_i, y_i) \mid \alpha_i \neq 0\}$$

(2) $\Rightarrow$ $$w = \sum_{\alpha_i \neq 0} \alpha_i y_i x_i$$

$$w = \sum_{x_i, y_i \in SV} \alpha_i y_i x_i$$

slope of HP is found from SV

for datapoint $(x_k, y_k)$

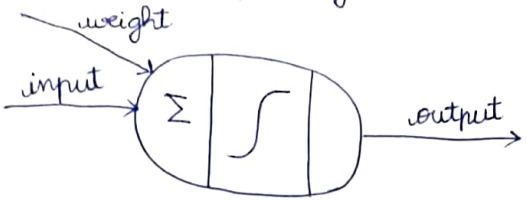
$$y_k = w x_k + b$$

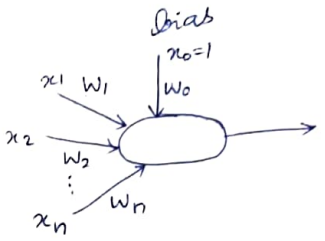
$$b = y_k - w x_k$$

* "linear SVM is called hard margin"

## PERCEPTRON :

- artificial neuron
- simulation of single neuron




weighted sum = $\sum_{j=0}^{d} x_j w_j$

activation function (brings non-linearity) = $\begin{cases} 1 & \text{if } \cos WS \geq \theta \\ 0 & \text{otherwise} \end{cases}$

MODEL : $W^T X$ PARAMETER : $W$ (for linearly separable)

- Sigmoid function - binary classification
- Softmax function - multi-way classification

## PERCEPTRON AS CONVEX OPTIMISATION PROBLEM:

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

find $W$

$$\min \quad SSE = \frac{1}{2} \sum_{i=1}^{m} (y_i - \hat{y}_i)^2$$

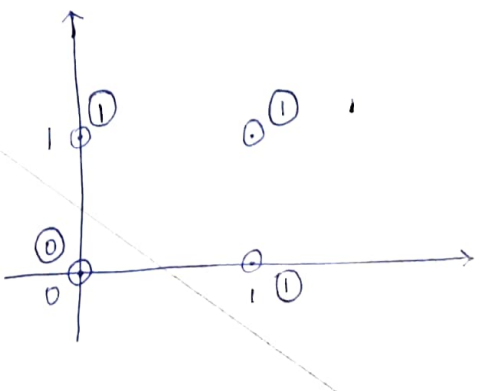
gradient $\frac{\partial SSE}{\partial W} = \sum (y_i - W^T x_i)(-x_i)$

forward pass : predicting $\hat{y}$
backward pass : updating $w$
} all deep learning algo follows

eg: Find $W$ for initial values $(0.5, 0.5, 0.5)$. $\eta = 1$
assume activation function is threshold function.
threshold value = 0.5

OR GATE

| | | |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |



EPOCH-1 $X = [0, 0, 1, 1]$ $[0, 1, 0, 1]$ $Y = [0, 1, 1, 1]$ $B = [0.5\ 0.5\ 0.5]$

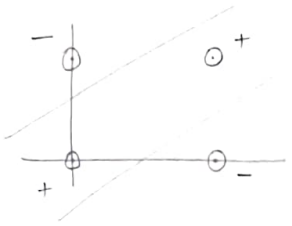
$B_n = B_0 + \eta$ grad

| $x_0$ | $x_1$ | $x_2$ | $b_0$ | $b_1$ | $b_2$ | $Y$ | $\hat{Y}$ | grad |
|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 0.5 | 0.5 | 0.5 | 0 | 0.5 >= 0.5 | [-1, 0, 0] |
| 1 | 0 | 1 | -0.5 | 0.5 | 0.5 | 1 | 0 | [1, 0, 1] |
| 1 | 1 | 0 | 0.5 | 0.5 | 1.5 | 1 | 1 | [0, 0, 0] |
| 1 | 1 | 1 | 0.5 | 0.5 | 1.5 | 1 | 1 (2.5 >= 0.5) | [ |

EPOCH-4 converged

XOR is not linearly separable. ∴ perceptron not applied.

| | | |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |




(MLP in PPT, refer gmail)

## SOFTMARGIN SVM :

$$\min \ \tfrac{1}{2}\|w\|^2 + C\sum_{i=1}^{n} I(\xi_i > 0)$$

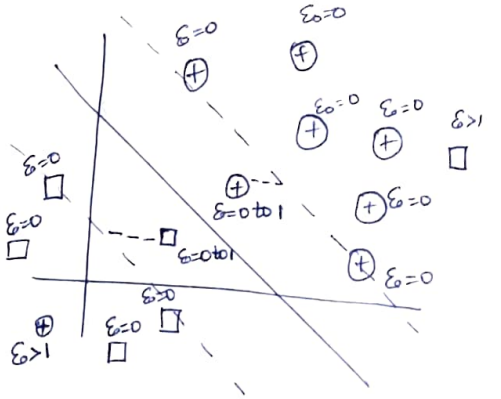
s.t.c

$$y_i(w^T x_i + b) + \underbrace{\xi_i}_{\text{slack variable}} \geq 1$$

where

C is the penalty

$$I(\xi > 0) = \begin{cases} 1 & \text{if } \xi > 0 \\ 0 & \text{if } \xi \leq 0 \end{cases}$$




* If C is high, number of errors will be reduced
* If $\xi_i = 0$, datapoints are correctly lied. Either lies on the margin or above.

For a □ datapoint

$$wx_i + b \leq -1$$

$$\Downarrow$$

$$wx_i + b + \xi_i \leq -1$$

For a ⊕ datapoint

$$wx_i + b \geq 1$$

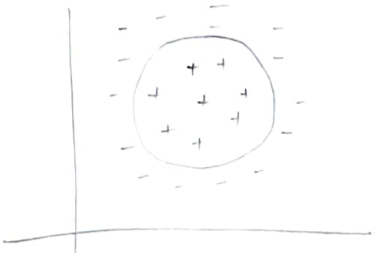
$$\Downarrow$$

$$wx_i + b + \xi_i \geq 1$$

$\xi_i = 0$ correct side

$0 \leq \xi_i \leq 1$ within the margin

$\xi_i > 1$ wrong side

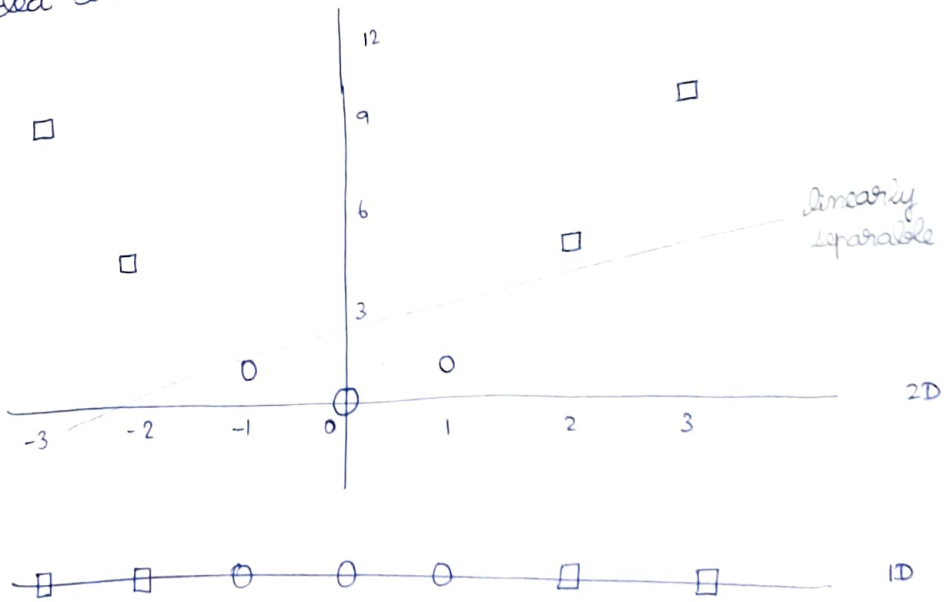


linearly non-separable problem

## SVM WITH KERNEL FUNCTION

A linearly non-separable problem when projected in higher dimension becomes linearly separable.

"Idea behind kernel SVM."




while multiplying $x_i$ $x_j$ vectors, increase dimension by applying kernel function. $K(x_i)$ $K(x_j)$

## KERNEL FUNCTIONS:

- linear
- polynomial $(x^T \cdot y)^d$ (d changes according to dim)
- gaussian
- radial basis function (RBF)
- exponential

eg: for dim = 2

$X = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$ $Y = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}$ $K(X, Y) = (x^T y)^2$

2 datapoints

kernel fn must be +ve semi-definite

$$x^T y = \begin{bmatrix} x_1 & x_2 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = x_1 y_1 + x_2 y_2$$

$$(x^T y)^2 = x_1^2 y_1^2 + x_2^2 y_2^2 + 2 x_1 x_2 y_1 y_2$$

$$\begin{bmatrix} x_1^2 \\ x_2^2 \\ \sqrt{2}\, x_1 x_2 \end{bmatrix} \qquad \begin{bmatrix} y_1^2 \\ y_2^2 \\ \sqrt{2}\, y_1 y_2 \end{bmatrix}$$

## UNSUPERVISED LEARNING

$$\mathcal{D} = \{ x_i \}_{i=1}^{n}$$

- clustering
- dimensionality reduction
- outlier analysis

## CLUSTERING:

$$\mathcal{D} = \{ x_i \}_{i=1}^{n}$$

$$\mathcal{D} = \{ C_1, C_2, C_3 \ldots C_k \}$$

$$C_k = \{ x_i \mid \text{distance}(x_i, \mu_k) = \text{minimum} \}$$

where $\mu_k$ is called cluster representative / centroid

## CLUSTERING ALGORITHMS

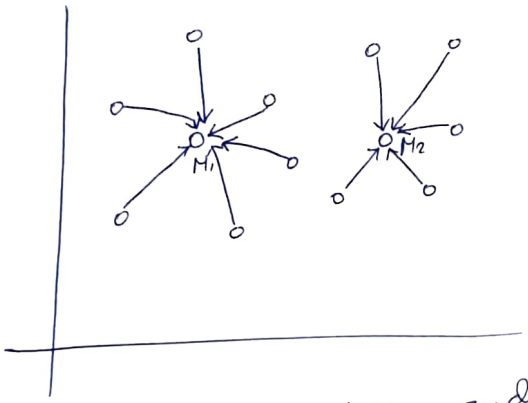
- partitioning
  - 1) k-means
  - 2) k-medoid
- hierarchical
  - topdown
    - 1) diana (divisive analysis)
  - bottom up
    - 1) agnes (agglomerative nesting)

## K-MEANS CLUSTERING :

input $\mathcal{D} = \{x_i\}_{i=1}^{n}$ & $K$

If you have domain knowledge, choose centroid.
Else choose any $K$ centroids randomly.

↳ sometimes min & max are chosen




Suppose $K = 2$

{ find distance metric
update centroid
↓
for each iteration

- Only in first iteration, a datapoint is centroid.
- Thereafter centroid is mean of all datapoints $\in$ clusters
- Successive 2 iterations if the centroid is same, optimality obtained.

eg: Apply K-means clustering and formulate 2 clusters. Use Euclidean distance as the distance metric. Choose centroids randomly.

| distance | speed |
|---|---|
| 71.24 | 28 |
| 52.53 | 25 |
| 64.54 | 27 |
| 55.69 | 22 |
| 54.58 | 25 |

$M_1 = (52.53, 25)$ min

$M_2 = (71.24, 28)$ max

Euclidean distance

| | $M_1$ | $M_2$ | |
|---|---|---|---|
| (64.54, 27) | 12.17 | 6.7 | $C_2$ |
| (55.69, 22) | 4.35 | 16.667 | $C_1$ |
| (54.58, 25) | 2.05 | 16.93 | $C_1$ |

To find optimum value of K, we use distortion as performance measure.

DISTORTION:

sum of squares of distances between datapoints & its corresponding centroid

$$\sum_{i=1}^{n} (x_i - \mu_i)^2$$

inertia → how well it is clustered for minimal K

Minimum the value of distortion, optimum the value of K

K-MEANS CLUSTERING:

unsupervised learning algorithm

Form $c_1, c_2 \ldots c_K$

which min distortion / intracluster distance (sum of all cluster distance)

$$\mathcal{D} = \sum_{x_i \in \mu_j} \| x_i - \mu_j \|^2$$

ALGORITHM: (time complexity : $O(nKd)$)

1. initialize the centroids randomly
2. while ! convergence
   a. for each $x_i \in \mathcal{D}$ // cluster formation

$$\mu_i = \underset{\mu_j}{\operatorname{argmin}} \| x_i - \mu_j \|^2 \qquad 1 \le j \le K$$

   b. for each cluster j // centroid updation

$$\mu_j = \frac{\sum_{x_i \in \mu_j} x_i}{n_j}$$

KMC is mixture of gaussian since each cluster follows normal dist. In ND, $\mu = \sigma$, it is sphere

LIMITATIONS:

* works only for spherical clusters
* convergence depends on selection of initial centroids

ASSUMPTION:

* closeness

## SPECTRAL CLUSTERING:

- based on spectral properties of graph
- given dataset $\mathcal{D}$ is represented as a graph $G$
  - * nodes - datapoints
  - * edges - connection between the nodes (affinity)
- $G$ may be directed / undirected
  - * adjacency matrix (complete graph)
  - * similarity matrix (use distance formula) (cosine similarity)
  - * KN-neighbour (connect only $k$ nearest neighbours)

} all are symmetric

eg: k means clustering dataset used

distance matrix

$$A = \begin{bmatrix} 0 & 18.95 & 6.77 & 16.67 & 16.93 \\ 18.95 & 0 & 12.98 & 4.36 & 2.05 \\ 6.77 & 12.18 & 0 & 10.16 & 11.14 \\ 16.67 & 4.36 & 10.16 & 0 & 3.199 \\ 16.93 & 2.05 & 11.14 & 3.199 & 0 \end{bmatrix}$$

$k=2$ nearest neighbours

$$\begin{bmatrix} 0 & 0 & 6.77 & 16.67 & 0 \\ 0 & 0 & 0 & 4.36 & 2.05 \\ 6.77 & 0 & 0 & 10.16 & 0 \\ 0 & 4.36 & 0 & 0 & 3.199 \\ 0 & 2.05 & 0 & 3.199 & 0 \end{bmatrix}$$

directed

degree matrix

$$D = \begin{bmatrix} 4 & 0 & 0 & 0 & 0 \\ 0 & 4 & 0 & 0 & 0 \\ 0 & 0 & 4 & 0 & 0 \\ 0 & 0 & 0 & 4 & 0 \\ 0 & 0 & 0 & 0 & 4 \end{bmatrix}$$

laplacian matrix

$L = D - A$

$$= \begin{bmatrix} 4 & -18.95 & -6.77 & -16.67 & -16.93 \\ -18.95 & 4 & -12.18 & -4.36 & -2.05 \\ -6.77 & -12.18 & 4 & -10.16 & -11.14 \\ -16.67 & -4.36 & -10.16 & 4 & -3.199 \\ -16.93 & -2.05 & -11.14 & -3.199 & 4 \end{bmatrix}$$

## PROPERTIES OF SPECTRUM:

* If there exists $d \times d$ symmetric matrix, then there will be $d$ eigen values. Every eigen value has its corresponding eigen vector. All eigen values are real and All eigen vectors are real and orthogonal.

* Sum of eigen values of covar matrix will give total variance.

$$\lambda_1 + \lambda_2 + \lambda_3 \ldots + \lambda_d = \sum_{j=1}^{d} \sigma_j^2$$

* Direction of maximum eigen value's eigen vector tells direction of variance.

## ALGORITHM:

input: dataset

output: clusters

1. formulate graph A
2. find degree matrix D as a diagonal matrix in which diagonals are degree
3. find laplacian L: D - A
   in which diagonals → degree
   off diagonals → negative edge weights
4. find eigen values of L
   # zero eigen values = # connected components / clusters
   first non-zero eigen value = fiedler value
5. find the eigen vector V for the fiedler value
6. for each value $x_m$ of eigen vector V
   a. if $x_m > 0$
      $x_m \rightarrow C_1$
   b. if $x_m < 0$
      $x_m \rightarrow C_2$

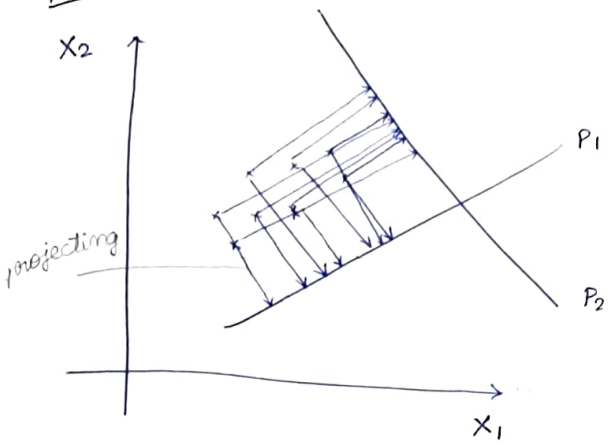
graph cut - fiedler value
no. of edges to be removed to make connected components

fiedler value: 2nd smallest eigen value of graph

## DIMENSIONALITY REDUCTION:

- principal component analysis (PCA) → unsupervised
- linear discriminant analysis (LDA) → supervised

### PCA:




If there exists 2 eigen vectors $P_1$ and $P_2$, choose vector which has more variance.
∵ data has to be visible. (scattered more)
$P_1$

### ALGORITHM:

input : $X = \{x_1, x_2, \ldots x_n\}$

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

$z = W^T x$

$z \in \mathbb{R}^k$ where $k < d$

1. $M = \{M_1, M_2, \ldots M_d\}$
2. mean subtracted data $X' = X - M$ ($M$ will be 0)
3. calculate covariance matrix (for mean subtracted data) ∵ $M = 0$

$$\Sigma_{ij} = \frac{\sum_{i=1}^{n} (x_i - M_j)(x_j - M_j)}{n-1} = \frac{\sum_{k=1}^{n} x_{ki} x_{kj}}{n-1}$$

4. eigen values of $\Sigma$, $\lambda_1, \lambda_2, \ldots \lambda_d$
5. arrange the eigen values in descending order
6. choose the largest eigen value ($\lambda_1, \lambda_2 \ldots \lambda_d$)
7. find the corresponding eigen vector EV.

$W_1, W_2 \ldots W_k$

8. $z = (W^T x)^T$ (k×n = k×d d×n)  $Z^T = X^T W$ (n×k = n×d d×k)

top k eigen vectors are principal components (of covariance matrix)

| | $X_1$ | $X_2$ |
|---|---|---|
| $P_1$ | 2 | 6 |
| $P_2$ | 1 | 7 |

| | $X_1$ | $X_2$ |
|---|---|---|
| $X_1$ | 0.5 | -0.5 |
| $X_2$ | -0.5 | 0.5 |

$\mu$ 1.5 6.5

$$\Sigma_{2\times 2} = \begin{array}{c} \\ X_1 \\ X_2 \end{array} \begin{array}{c} \begin{array}{cc} X_1 & X_2 \end{array} \\ \begin{bmatrix} 0.5 & -0.5 \\ -0.5 & 0.5 \end{bmatrix} \end{array}$$

$$|\Sigma - \lambda I| = 0$$

$$\Sigma - \lambda I = \begin{bmatrix} 0.5 & -0.5 \\ -0.5 & 0.5 \end{bmatrix} - \begin{bmatrix} \lambda & 0 \\ 0 & \lambda \end{bmatrix}$$

$$= \begin{bmatrix} 0.5-\lambda & -0.5 \\ -0.5 & 0.5-\lambda \end{bmatrix}$$

$$|\Sigma - \lambda I| = 0$$

$$(0.5-\lambda)(0.5-\lambda) - (-0.5)(-0.5) = 0$$

$$0.25 + \lambda^2 - \lambda - 0.25 = 0$$

$$\lambda(\lambda - 1) = 0$$

$\boxed{\lambda = 0}$ $\boxed{\lambda = 1}$ eigen values

finding eigen vector for $\lambda = 1$ (max)

$$\Sigma - \lambda I = \begin{bmatrix} 0.5-1 & -0.5 \\ -0.5 & 0.5-1 \end{bmatrix} = \begin{bmatrix} -0.5 & -0.5 \\ -0.5 & -0.5 \end{bmatrix}$$

to form characteristic equation

$$\begin{bmatrix} -0.5 & -0.5 \\ -0.5 & -0.5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = 0$$

$$-0.5x_1 - 0.5x_2 = 0$$

$$-0.5x_1 - 0.5x_2 = 0$$

$$-0.5x_1 = 0.5x_2$$

$$-x_1 = x_2$$

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

to convert into unit vector divide by norm

$$\text{norm} = \sqrt{2} \quad \left(\sqrt{1^2 + (-1)^2}\right)$$

$$W = \left(\frac{1}{\sqrt{2}}, \frac{-1}{\sqrt{2}}\right)$$

unit vectors act like axis

$$z = W^T X$$

$$= \begin{bmatrix} 1/\sqrt{2}, & -1/\sqrt{2} \end{bmatrix} \begin{bmatrix} 0.5 & -0.5 \\ -0.5 & 0.5 \end{bmatrix}$$

$$z = \begin{bmatrix} 1/\sqrt{2} & -1/\sqrt{2} \end{bmatrix}$$

+ into unit vector divide by norm

+

20PW13

**PSG COLLEGE OF TECHNOLOGY, COIMBATORE - 641 004**

**Department of Applied Mathematics and Computational Sciences**

**M.Sc. (SS) – V Semester**

**TEST - I**

**20XW53 - MACHINE LEARNING**

**Time: 1 Hour 15 min.** **Maximum Marks: 40**

**INSTRUCTIONS:**

1. Answer **ALL** questions. Each question carries 20 Marks.
2. Subdivision (a) has 2 questions, each carries 2 marks, subdivision (b) carries 6 marks each and subdivision (c) carries 10 marks each.

1. a) i. Check the convexity of cross entropy loss $L = y_i \log \hat{y}_i$ where $\hat{y}_i = W^T X$. (2) concave

ii. Differentiate generative and discriminative models. (2)

b) Define parameters and hyperparameters of a model. State the parameters and hyperparameters (if any) for the following ML algorithms? i. Polynomial regression ii. Multiple regression iii. Ridge regularization. Elaborate a method on finding the optimal value of hyperparameters? 3 way split

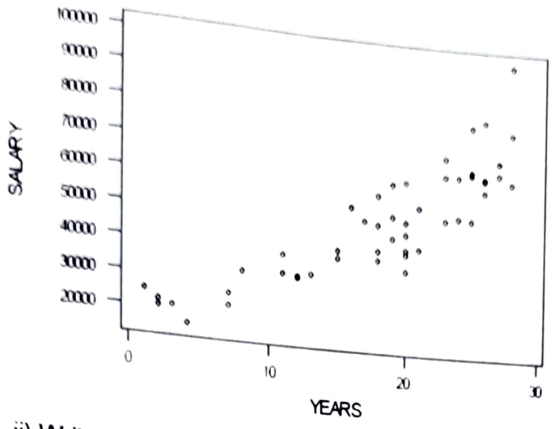
c) What are the characteristics of gradient descent method? Design an algorithm for finding a solution of an optimization problem. Explain its variants with procedure. What are the possible criteria for convergence in Gradient descent (GD) method? Justify the convergence mathematically. Given a data set with 2000 instances and 10 features, how many times parameter will be updated in an epoch when the dataset is applied on GD and its variants?

2. a) i. The scatter plot shows the employees' years of experience vs their salary for a sample of 50 managers. The regression model obtained from data is salary = 11369+2141* years and what is the sum of residuals (errors) of the model?

$SE = \sum (y_i - \hat{y}_i) = 0$ for linear regression (2)

to convert into unit vector divide by norm

$\frac{1}{\|x\|}(x_1, x_2, x_3)$




ii) Write an unconstrained optimization problem for Ridge regularization and a solution to the problem. $\sum_{i=1}^{n}(y_i - w^T x_i)^2 + \lambda \sum_{j=1}^{d}(w_j^2)$ $W = (X^T X + \lambda I)^{-1} X^T y$

b) What are the inferences about the model (M) and data (D) when M overfits D. Elaborate the impact of following factors in Multiple regression. How do you alleviate these concerns in an ML model?

i. Multicollinearity

ii. Outliers

Assume that a regression data set D contains 100 independent features and a dependant variable. A ML engineer who has no knowledge about the relevance and importance of these features influencing dependant variable, applied Multiple Regression on data set D. He encountered an overfitting issue. Which technique do you suggest him to resolve the overfitting? Why? Give reasons.

Lasso regression

c) Given a data set with a continuous independent variable X, a dependant variable Y, construct a quadratic polynomial regression model and answer the following:

Infer the relationship between dependant and independent variables from the parameters. Obtain the irreducible error of the model and find the proportion of variance explained by X to determine Y. SSR

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| X | 1 | 2 | 3 | 4 | 4 | 5 | 5 | 6 | 6 | 7 |
| Y | 7 | 8 | 9 | 8 | 9 | 11 | 10 | 13 | 14 | 13 |

**********

20PW13

# PSG COLLEGE OF TECHNOLOGY, COIMBATORE - 641 004

## Department of Applied Mathematics and Computational Sciences

**M.Sc. (SS) – V Semester**

**TEST - II**

**20XW53 - MACHINE LEARNING**

**Time: 1 Hour 15 min.** **Maximum Marks: 40**

**INSTRUCTIONS:**

1. Answer **ALL** questions. Each question carries 20 Marks.
2. Subdivision (a) has 2 questions, each carries 2 marks, subdivision (b) carries 6 marks each and subdivision (c) carries 10 marks each.

1. a) i. Given a dataset D with n observations and each observation consists of an input vector $x_i \in R^d$ *and an outcome* $y_i \in \{c1, c2, \ldots, cm\}$ , infer the information, purity, surprisal of Y when the Entropy H(Y) is $\log_2 m$ ? What is bias in decision tree? (2)

ii. Differentiate lazy learners with eager learners. (2)

b) Prove that logistic regression is a linear model. Let us assume that a binomial logistic regression is applied on a dataset with an input entrance marks, and a class variable with values pass or fail. The model parameters obtained are w0 = 1, w1 = 8. Compute the prediction for a student with entrance marks is 70.

c) Apply Naïve Bayes classifier for the below data and construct a model. Assume that taxable income is a continuous feature and normalize it by dividing the value by 1000. Do smoothing in case of zero frequency problem. Also predict test data with values <No, married, 120K>

| *Tid* | Refund | Marital Status | Taxable Income | Evade |
|---|---|---|---|---|
| 1 | Yes | Single | 125K | No |
| 2 | No | Married | 100K | No |
| 3 | No | Single | 70K | No |
| 4 | Yes | Married | 120K | No |
| 5 | No | Divorced | 95K | Yes |
| 6 | No | Married | 60K | No |
| 7 | Yes | Divorced | 220K | No |
| 8 | No | Single | 85K | Yes |
| 9 | No | Married | 75K | No |
| 10 | No | Single | 90K | Yes |

P(Y) =

to convert into unit vector divide by norm

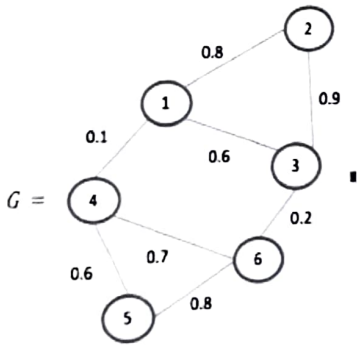
norm $\sqrt{\quad}$ $/\ \sqrt{1^2+(-1)^2}$ )

2. a) i. Is it possible to apply k-means clustering for clusters of non-convex sets? Justify your answer.

ii. Suppose we have three cluster centroids C1 = (1, 2), C2 = (-3, 0) C3 = (4.2). Furthermore, we have a training example X = (-1,2). Based on k-means clustering with Euclidean distance, X will be assigned to which cluster?

b) Consider a hyperplane function of Linear SVM for two variables is $w_1x_1 + w_2x_2 + b$ . If two hyperplanes considered in a hypotheses space are $2x_1 + 3x_2 + 4$ and $10x_1 + 3x_2 + 1$ Find the cost or objective function for each hyperplane and pick an optimum hyperplane. Write the optimization problem for soft margin SVM. What is the impact of hyper-parameters in it?

c) How is spectral clustering advantageous to k-means clustering? List the steps in Spectral clustering. Below graph G gives the representation of a dataset with six data points.




Obtain the Laplacian matrix L from the above weighted undirected graph. List the characteristics of Laplacian matrices. Infer the possible Eigen values. What would be the possible Fiedler value for L? Find the sum of all Eigen values. Assume that Fiedler vector is [ -0.4 -0.3 -0.5 0.6 0.7 0.8] . How do you form clusters? Justify your answer wherever required.

**********