12/07

→ In traditional programming, we define the problem statement clearly, find the solution, determine the factors that determine the output (i.e, inputs), implement it through code and test it to obtain a better program

→ Unlike the traditional programming, data acts as an input and there is no requirement of explicit programming

→ Artificial Intelligence - making the machine think, giving intelligence to the machine

→ We have to provide the machine with the knowledge base so that it comes to know the possible moves to be made.

→ For developing the knowledge base, a domain expert is required to define because the knowledge of the entire domain is required.

→ Example, in a chess player scenario, we have to have all the possible moves, for assassination of the opponent, movement of each, etc.

→ ML makes the machine learn from the enormous amount of data.

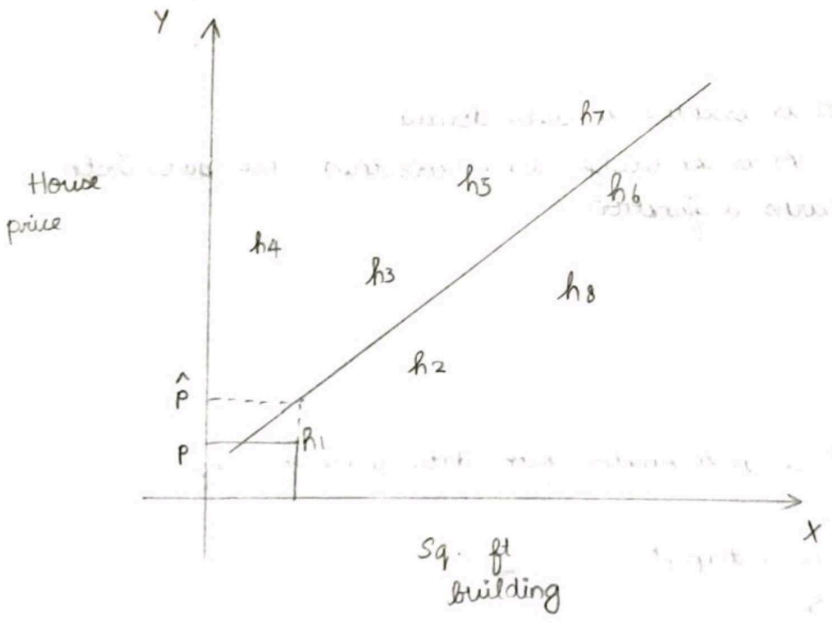
→ One problem statement example is the ham and spam mails. We need certain ham and spam mails that serve as the data.

→ Then, it has to be differentiated. ML algorithm does this

→ The main objective of a ML algorithm is to find a function that maps an input to an output.

$f \quad X \rightarrow Y$ (output)

(input)



→ Linear regression tries to find a straight line that fits in most of the points

→ p is the actual price, $\hat{p}$ is the approximated (predicted) price by the ML algorithm

$p - \hat{p}$ is the error, it may be positive or negative.

→ In order to make it a positive quantity we square it and call it the squared error.

→ Sum of Squared Errors (SSE) is given by $\sum_{i=1}^{8} (p_i - \hat{p}_i)^2$

→ We have to find the straight line that minimises the above stated error and as the standard straight line equation goes

$y = mx + c$ where m and c are parameters

→ The line equation is the function or the model in this example.

→ The above stated example is the least squares method using linear regression

→ The ML algorithm constructs a model from the given data. Representing it through conditional probability

model / data.

→ ML algorithms have their own assumptions. Ex: As X increases, Y also increases in the above example.

→ In the example, details about various houses serve as the data.

→ If we provide data to a MLA, we get a function as an output which is equivalent to a program in the traditional programming.

→ Building an expert system is essential in every domain.

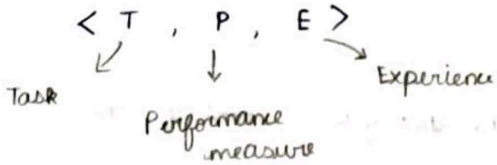
→ ML is a subset of AI. AI is knowledge - base concerned. ML gives both input and output and it returns a function.

18/07

Machine learning :

→ Study of algorithms that learn a function from data without being explicitly programmed

→ Tom Mitchell defined ML as a triplet



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

Terminologies in ML:

1. Hypothesis: It is the assumptions about the function / model. It is simply the function / model / concept 'h'

ML algorithms	'h'	'H'
1. Linear regression	- straight line equation	$H = \{ mx + c \mid m, c \text{ are real} \}$
2. Polynomial regression	- polynomial of degree d	
3. Naive Bayes	- probabilistic	
4. K - Nearest Neighbours	- k closest neighbour of a data point	
5. Support Vector Machines		
6. Decision trees		

2. Hypothesis space: 'H'. Set of all possible 'h', where 'h' is the hypothesis

$H = \{ h \mid h \text{ is } h(x) \}$

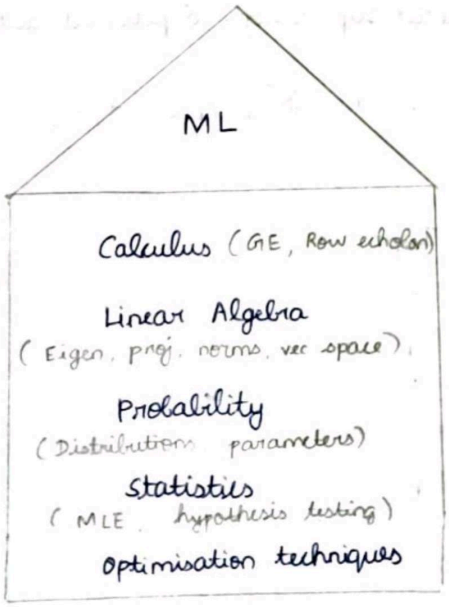
Finding an optimal hypothesis which minimises the error is needed.

3. Parameters: The unknown values in the hypothesis. They are obtained by solving the optimisation problem.

$f(x) = w_0 + w_1 x + w_2 x^2 + \dots + w_d x^d$

Here, $w_0, w_1, \dots w_d$ are parameters.

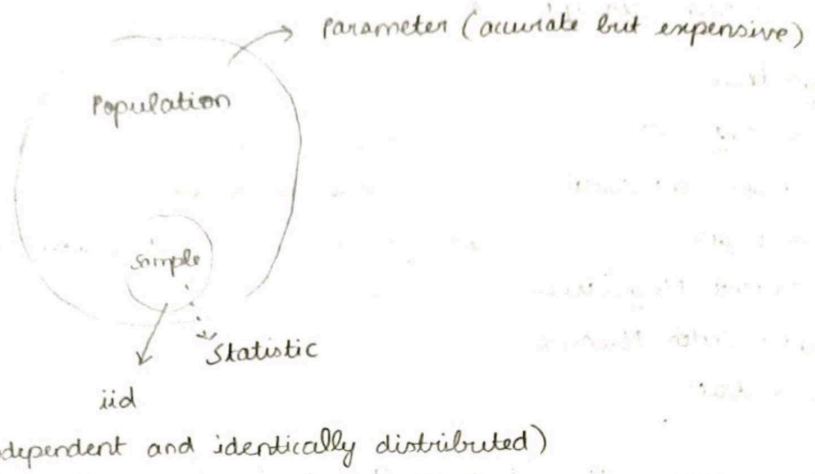
4. Hyper parameters: Parameters that are used in training algorithms.
Ex: Degree of the polynomial determination



→ Random variable is a function that maps all possible outcomes to some real number. Types are discrete and continuous.

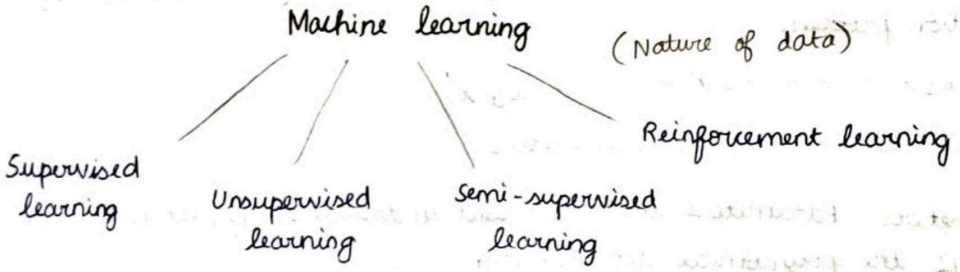
→ When the random variable takes only 2 values, then it is called Bernoulli distribution. Parameters are p and q

→ Most of the continuous random variables follow Gaussian / Normal distribution which is bell-shaped. Parameters are μ and σ.



→ Statistic is estimator for population. ML does that. MLE is a technique which is used

19/07



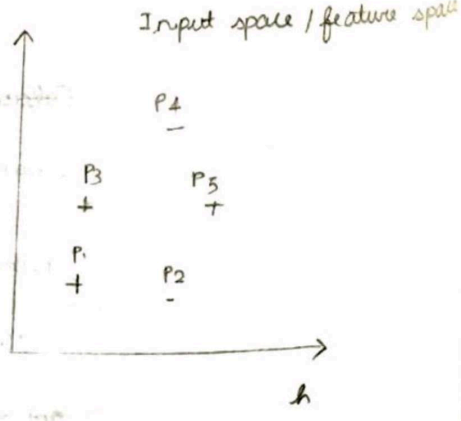
Supervised learning : Learning under supervision. Supervised data.

Data set · $D = \{(x_i, y_i)\}_{i=1}^{n}$, $x_i \in \mathbb{R}^2$, $y_i \in \{+, -\}$

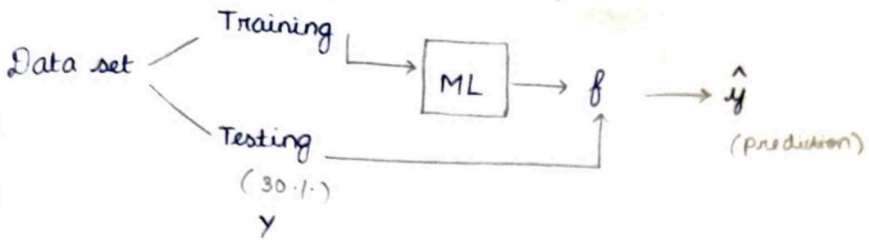
i/p / feature / characteristic / attribute

o/p, o/c or class label

Height	Weight	Type
5	60	C (+)
6	70	F (-)
6.1	80	C
5.7		F
		C



Data set is considered as a matrix where rows are datapoints / observations / instances / examples and columns are features / attributes / characteristics / input vector



Since the data set has both input and output, it is called supervised learning. The training data is given to the ML algorithm to obtain the function / model which is in turn evaluated using the test data.

Supervised learning

o/p variable based

Classification
- Y i.e. the o/p variable is discrete.
- Ex: Spam mail detection

Regression
- The o/p variable is continuous
- Ex: Rainfall prediction

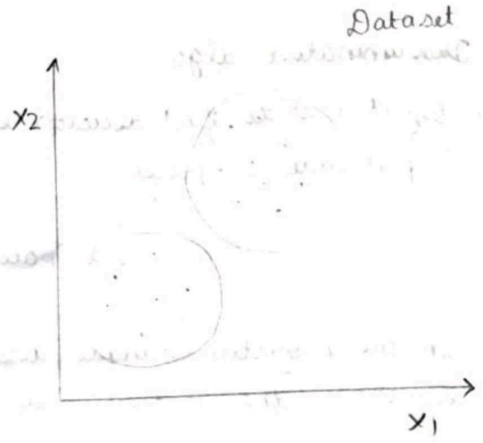
Classification Algorithms
1. Naive Bayes
2. Logistic regression
3. SVM
4. KNN
5. Linear Discriminant Analysis (LDA)
6. QDA
7. Decision tree
8. Random forest

Regression Algorithms
1. Linear regression
2. Polynomial regression
3. Multiple regression
4. Auto regression
5. KNN
6. SVM
7. Decision tree

Few classification algorithms are used for regression too.

Unsupervised learning

→ Helpful in finding hidden patterns.
→ Data set has only input features and not the class labels
→ Grouping the data points into clusters.
→ Data points in the same cluster share few common characteristics.



Unsupervised learning involves:
* Clustering
* Dimensionality reduction (100 - 2)

For better visualisation and to avoid redundant data, eliminate irrelevant features

* Outlier Analysis

If you include these pts in the data set, it will affect the accuracy of the model and hence we need to remove

25/07

Semi-supervised learning

→ Only few labelled datapoints, apply supervised learning. Obtain a model, predict for unlabelled datapoints.

→ Since the model is not good (because of very few labelled datapoints), we obtain the confidence level for the predictions.

→ We put the 5 most confident points into the training set so that the set becomes slightly larger.

Reinforcement learning.

→ Ex: Robot training, game playing.

→ Consider a game, sequence of moves, we want the system to make a move, based on the quality of the move, we give either a reward or a penalty.

→ Our main objective here is to maximise the reward / minimise the penalty.

→ In case of robot movement, if it encounters an obstacle, we give a penalty otherwise a reward. (No predefined training set)

Algorithms

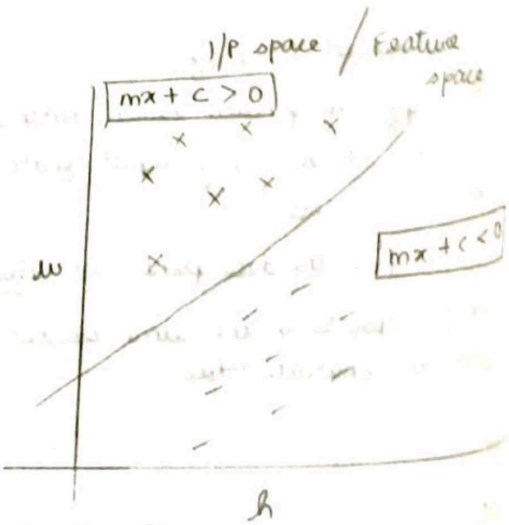
Q-learning algorithm

Machine learning algo. (based on models)

- Generative
- Discriminative

Discriminative algo.

→ Objective is to find decision boundary (split into 2 spaces) → subspaces actually!



In the projected example, using the line equation, the feature space is divided into 2 equal / unequal subspaces.

(Discriminate func.)

It need not be a line equation always. It may be a hyperbolic rep. too.

Examples: Perceptron, SVM, logistic regression.

→ It also tries to find posterior / conditional probability $P(y|x)$
(After analysing height & weight, we predict the nature of the player)

Generative algo.

→ Based on the learning, it tries to generate the data
→ Tries to learn the whole data space
→ Also finds the joint probability $P(x, y)$, $P(h, w, c)$

Zoo example - drawing the structure of zebra and elephant - differentiating both the animals based on their features

Examples. Naive Bayes, HMM

ML algorithms

(based on the parameters of the model)

Parametric Non-parametric

Parametric

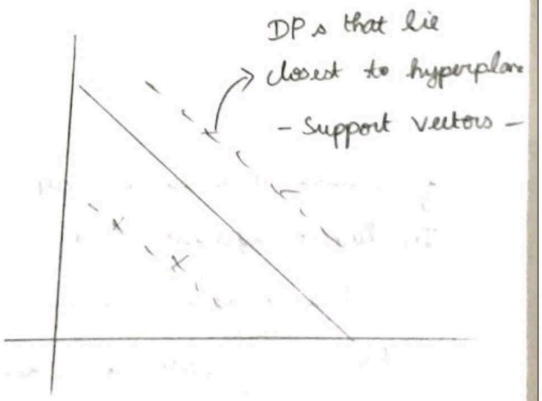
→ Tries to construct a model with fixed number of parameters irrespective of the size of the data set. The value of the parameters may vary.

Examples: Perceptron, SVM, Logistic regression

Non-parametric

→ No parameters, i.e, no fixed set of parameters / no parameters itself.
→ Decision boundary varies from data point to data point.
→ No. of parameters may vary.
→ Here, support vectors are varied as the size of the data increases / decreases.

Examples: SVM, KNN



In SVM there are 2 kinds
- Linear SVM (parametric)
- Non-linear SVM (non-parametric)

?

26/07

LINEAR REGRESSION

Given, (observations)

$$D = \{ x_i, y_i \}_{i=1}^{n} \quad , \quad x_i \in \mathbb{R} \, , \; y_i \in \mathbb{R} \, , \; f : x_i \to y_i$$

Objective:

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

Tries to find a linear function of independent and dependent variables.

$$Y = f(x)$$

Rainfall, temp., (ind
living area in sq.ft.

Consider a data set,

	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

f : $16x + 5$

$\hat{f}$: $8x + 5$

$\hat{f}$ because it is an estimator for the true function.

In linear regression (LR),

True function, $f : \beta_0 + \beta_1 x$ (population parameters)

Predicted function, $\hat{f} : b_0 + b_1 x_1$ (estimators), $b_0 \approx \beta_0$ and $b_1 \approx \beta_1$

> When not to apply linear regression?

When the variables are not linearly related.

The value of Y may not depend only on the x, it may have different other factors which influence it.

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

y : $\underbrace{b_0 + b_1 x}_{\hat{y}} + e$

Sum of Squared Errors (SSE) = $\sum_{i=1}^{n} (y_i - \hat{y}_i)^2$

Find b_0, b_1 such that it minimises the error. } → Optimisation pblm

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

$$= \sum_{i=1}^{n} (y_i - (b_0 + b_1 x_i))^2 \quad \text{[Substituting } \hat{y}_i \text{]}$$

$$\frac{\partial}{\partial b_0} \text{SSE} = 0 \quad , \quad \frac{\partial}{\partial b_1} \text{SSE} = 0$$

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

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

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

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

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

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

→ ①

Solving ① and ②: (Square matrix)

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

$$X^T Y = X^T X \cdot B$$

Multiply $(X^T X)^{-1}$ on both sides.

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

$$X = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_n \end{bmatrix}_{n\times 2}, \quad b = \begin{bmatrix} b_0 \\ b_1 \end{bmatrix}, \quad Y = \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{bmatrix}_{n\times 1}, \quad X^T = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ x_1 & x_2 & \cdots & x_n \end{bmatrix}_{2\times}$$

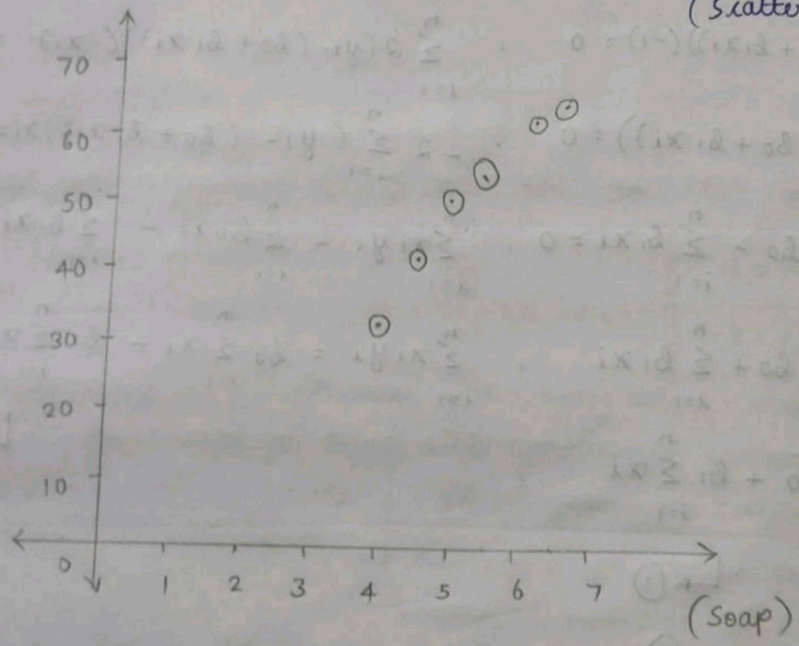
$$X^T X = \begin{bmatrix} n & \Sigma x_i \\ \Sigma x_i & \Sigma x_i^2 \end{bmatrix}$$

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

X has the first column as 1s because $X \cdot b$ gives the y function

x	Soap	4	4.5	5	5.5	6	6.5	7
Y	Sud	33	42	45	51	53	61	62

(Scattered plot)



$$X = \begin{bmatrix} 1 & 4 \\ 1 & 4.5 \\ 1 & 5 \\ 1 & 5.5 \\ 1 & 6 \\ 1 & 6.5 \\ 1 & 7 \end{bmatrix}, \quad X^T = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 4 & 4.5 & 5 & 5.5 & 6 & 6.5 & 7 \end{bmatrix}, \quad Y = \begin{bmatrix} 33 \\ 42 \\ 45 \\ 51 \\ 53 \\ 61 \\ 62 \end{bmatrix}$$

$$X^T X = \begin{bmatrix} 7 & 38.5 \\ 38.5 & 218.75 \end{bmatrix}$$

$$B = \begin{bmatrix} b_0 \\ b_1 \end{bmatrix} = (X^T X)^{-1} X^T Y$$

$$(X^T X)^{-1} = \begin{bmatrix} 4.464 & -0.785 \\ -0.785 & 0.143 \end{bmatrix}$$

$$X^T Y = \begin{bmatrix} 347 \\ 1975 \end{bmatrix}$$

$$B = \begin{bmatrix} 4.464 & -0.785 \\ -0.785 & 0.143 \end{bmatrix} \begin{bmatrix} 347 \\ 1975 \end{bmatrix} = \begin{bmatrix} 1531.6580 - 1550.375 \\ -272.395 + 282.425 \end{bmatrix} = \begin{bmatrix} -18.5 \\ 80.1 \end{bmatrix} \begin{matrix} -2.6786 \\ 9.5 \end{matrix}$$

Predict the sud value for soap value 4.25

$$y = b_0 + b_1 x$$

$$y = -2.6786 + 9.5(4.25)$$

$$y = -2.6786 + 40.375$$

$$y = 37.6964$$

→ Sum of errors / mean of errors of linear regression is zero.

Evaluation measures :

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

2. Sum of Squared Errors (SSE) = $\sum_{i=1}^{n} (y_i - \hat{y}_i)^2$ $\qquad SST = SSE + SSR$

3. Regression Sum of Squares (SSR) = $\sum_{i=1}^{n} (\hat{y}_i - \bar{y})^2$

4. Coefficient of determination

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

$$y - \bar{y} = (y - \hat{y})^2 + (\hat{y} - \bar{y})$$

SST SSE SSR

27/07

→ Identify independent and dependent features i.e, variables in the house price prediction. Draw scattered plot for every ind. and dep. variable and interpret the relationship between the ind. and dep. variable. Then, construct a model using linear regression for every dep. and ind. feature and evaluate the models using different measures such as SSE, SST, SSR and coefficient of determination and find out the most important feature that influences the dependent variable. Find outliers if any.

a = [] [] a = [[ ]]

for i in (.then.

z = [1, list[i]]

a.append(z)

$(A-B)^2$

10	11	12
9	10	11

$(1)^2 + (1)^2 + (1)^2 = 3$

29/07

→ SST gives the total variation i.e, how far each value is deviated from the mean (or) how far the values of Y are spread.

→ SSR tells how the model explains the deviation. (solves the variance ?)

$$0 \leq SSR \leq SST$$

Larger value, the better model.

→ r^2 - how much error is explained by the model.

If $r^2 = 1$, that is considered the best model. // when SSR = SST

Calculate SST, SSE and SSR for soap and sud data set.

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

Predicting the values and tabulating it with actual values

Soap	4	4.5	5	5.5	6	6.5	7
Actual Sud	33	42	45	51	53	61	62
Predicted Sud	35.321	40.071	44.821	49.571	54.321	59.071	63.821

$$SST = \sum_{i=1}^{7} (y_i - \bar{y})^2$$

$274.598 + 57.320 + 20.8940 + 2.042 + 11.758 + 130.622 + 154.4800$

651.714

$$SSE = \sum_{i=1}^{7} (y_i - \hat{y}_i)^2$$

$5.387 + 3.721 + 0.032 + 2.042 + 1.745 + 3.721 + 3.316$

19.964

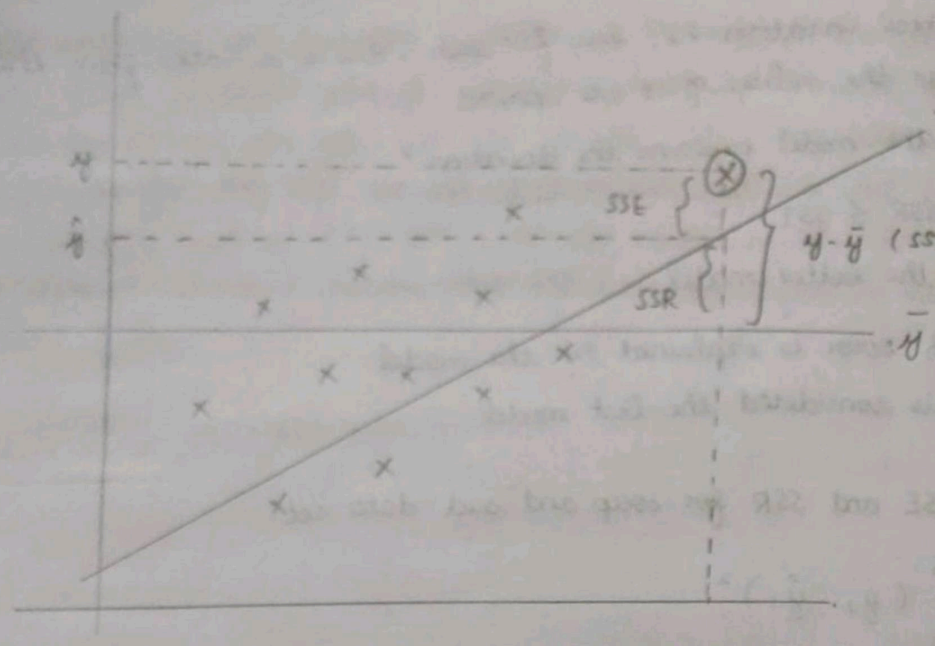
$SSR = SST - SSE$ $\quad 203.0625 + 90.25 + 22.5625 + 0 + 22.5625 + 90.25 + 203.0625$

$SSR = 631.75$

$$r^2 = \frac{SSR}{SST} = \frac{631.75}{651.714} = 0.9693$$

Coefficient of correlation, $r = \frac{cov(x,y)}{\sigma_x \sigma_y}$, $\frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{(n-1)\sigma_x \sigma_y}$

$$\sigma_x = \sqrt{\frac{\sum (x_i - \bar{x})^2}{n-1}} \; 5.5, \quad \sigma_y = \sqrt{\frac{\sum (y_i - \bar{y})^2}{n-1}} \; 49.571$$



$$\sigma_x^2 = \frac{2.25 + 1 + 0.25 + 0 + 0.25 + 1 + 2.25}{6}$$

$$\sigma_x^2 = \frac{7}{6} = 1.17$$

$$\sigma_x = 1.0817$$

$$\sigma_y^2 = \frac{274.598 + 57.32 + 20.894 + 2.042 + 11.758 + 130.622 + 154.48}{6}$$

$$\sigma_y^2 = \frac{651.714}{6} = 108.619$$

$$\sigma_y = 10.422$$

$$\rho = \frac{617.8455 + 57.32 + 5.2235 + 0 + 2.9395 + 130.622 + 347.58}{6(1.0817)(10.422)}$$

$$\rho = \frac{1161.5305}{67.6409} = 17.1720$$

$$\rho = \frac{1.5(16.57) + 1(7.571) + 0.5(4.571) + 0 + 0.5(3.429) + 1(11.429) + 1.5(12.429)}{67.6409}$$

$$= \frac{24.8565 + 7.571 + 2.2855 + 0 + 1.7145 + 11.429 + 18.6435}{67.6409}$$

$$\rho = 0.9831$$

HW

Construct a simple LR model b/w sepal length and sepal width, petal length and petal width. Calculate evaluation metrics. Consider 6 rows from the start.

Sepal width (x)	3.5	3	3.2	3.1	3.6	3.9
Sepal length (y)	5.1	4.9	4.7	4.6	5	5.4

$$X = \begin{bmatrix} 1 & 3.5 \\ 1 & 3 \\ 1 & 3.2 \\ 1 & 3.1 \\ 1 & 3.6 \\ 1 & 3.9 \end{bmatrix}, \quad X^T = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 3.5 & 3 & 3.2 & 3.1 & 3.6 & 3.9 \end{bmatrix}, \quad y^T = \begin{bmatrix} 5.1 \\ 4.9 \\ 4.7 \\ 4.6 \\ 5 \\ 5.4 \end{bmatrix}$$

$$X^TX = \begin{bmatrix} 6 & 20.3 \\ 20.3 & 69.27 \end{bmatrix}, \quad X^TY = \begin{bmatrix} 29.7 \\ 100.91 \end{bmatrix}$$

$$(X^TX)^{-1} = \begin{bmatrix} 19.623 & -5.75 \\ -5.75 & 1.6997 \end{bmatrix}$$

$$B = (X^TX)^{-1} X^TY$$

$$B = \begin{bmatrix} 19.623 & -5.75 \\ -5.75 & 1.6997 \end{bmatrix} \begin{bmatrix} 29.7 \\ 100.91 \end{bmatrix} = \begin{bmatrix} 2.5706 \\ 0.7417 \end{bmatrix}$$

The constructed model is $\hat{y} = 2.5706 + 0.7417(x)$

Including the predicted values in the table,

Sepal width (x)	3.5	3	3.2	3.1	3.6	3.9
Sepal length (y)	5.1	4.9	4.7	4.6	5	5.4
Predicted, $\hat{y}$	5.16655	4.7957	4.94404	4.86987	5.24072	5.46323

$$SSE = \sum_{i=1}^{6} (y_i - \hat{y}_i)^2 = 0.00443 + 0.01088 + 0.05955 + 0.07283 + 0.05795 + 0.00399$$
$$= 0.20963$$

$$SST = \sum_{i=1}^{6} (y_i - \bar{y})^2 = 0.0225 + 0.0025 + 0.0625 + 0.1225 + 0.0025 + 0.2025$$
$$= 0.415$$

$$SSR = SST - SSE = 0.20537 \qquad 0.42508$$

$$r^2 = \frac{SSR}{SST} = 0.49487, \quad \sqrt{r^2} = \rho = 0.70347$$

Petal width (x)	0.2	0.2	0.2	0.2	0.2	0.4
Petal length (y)	1.4	1.4	1.3	1.5	1.4	1.7

$$X = \begin{bmatrix} 1 & 0.2 \\ 1 & 0.2 \\ 1 & 0.2 \\ 1 & 0.2 \\ 1 & 0.2 \\ 1 & 0.4 \end{bmatrix}, \quad X^T = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 0.2 & 0.2 & 0.2 & 0.2 & 0.2 & 0.4 \end{bmatrix}, \quad Y = \begin{bmatrix} 1.4 \\ 1.4 \\ 1.3 \\ 1.5 \\ 1.4 \\ 1.7 \end{bmatrix}$$

$$X^TX = \begin{bmatrix} 6 & 1.4 \\ 1.4 & 0.36 \end{bmatrix}, \quad X^TY = \begin{bmatrix} 8.7 \\ 2.08 \end{bmatrix}$$

$$(X^TX)^{-1} = \begin{bmatrix} 1.8 & -7 \\ -7 & 30 \end{bmatrix}$$

$$B = (X^TX)^{-1}X^TY = \begin{bmatrix} 1.8 & -7 \\ -7 & 30 \end{bmatrix}\begin{bmatrix} 8.7 \\ 2.08 \end{bmatrix} = \begin{bmatrix} 1.1 \\ 1.5 \end{bmatrix}$$

The predicted model is $1.1 + 1.5(x)$ i.e, $\hat{y}$.

Tabulating the predicted values

Petal width (x)	0.2	0.2	0.2	0.2	0.2	0.4
Petal length (y)	1.4	1.4	1.3	1.5	1.4	1.7
Predicted, $\hat{y}$	1.4	1.4	1.4	1.4	1.4	1.7

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

$$SSE = 0 + 0 + 0.01 + 0.01 + 0 + 0 = 0.02$$

$$SST = \sum_{i=1}^{6} (y_i - \bar{y})^2, \quad \bar{y} = 1.45$$

$$SST = 0.0025 + 0.0025 + 0.0225 + 0.0025 + 0.0025 + 0.0625$$

$$SST = 0.01 + 0.01 + \dots + 0.01 + 0.01 + 0.095$$

$$SSR = SST - SSE = 0.095 - 0.02 = 0.075$$

$$r^2 = \frac{SSR}{SST} = \frac{0.075}{0.095} = 0.78947$$

$$\sqrt{r^2} = \rho = 0.88852$$

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

- There may be matrices that are not invertible. In those cases inverse can't be computed and in further model cannot be constructed.
- Also, when the size of the data set ↑, the size of matrix X i.e, $m \times 2$ increases and the time complexity increases.

Because of this, we go to better models.

Gradient descent method

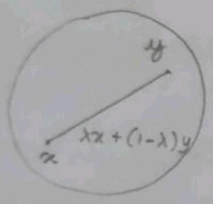
* Iterative
* First order derivative

Pre-req

- Unconstrained convex opt. prob.

Convex set

$0 \le \lambda \le 1$

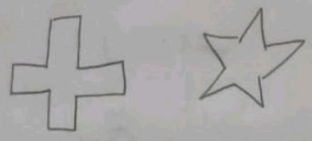


$C \in \mathbb{R}^2$

$x\lambda + (1-\lambda)y \in C, \quad 0 \le \lambda \le 1$

Eg:

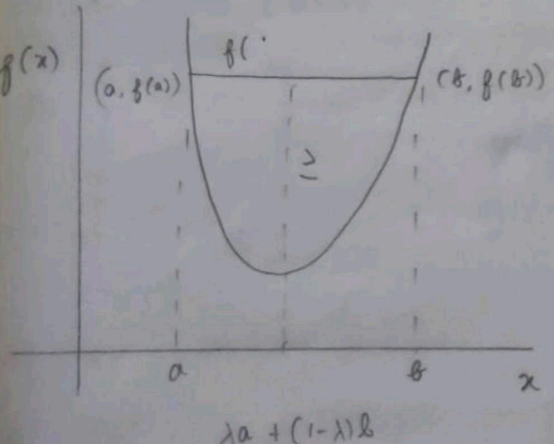
Convex sets



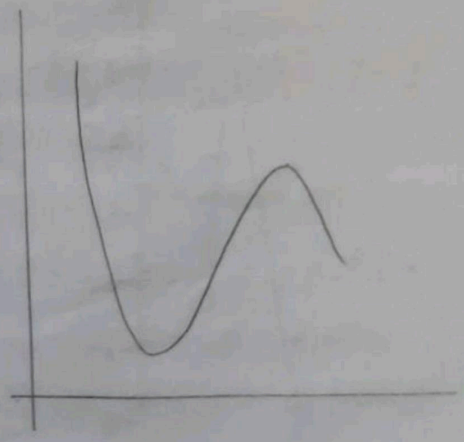
Non-convex sets

Intersection of two convex sets is a convex set.

Convex function

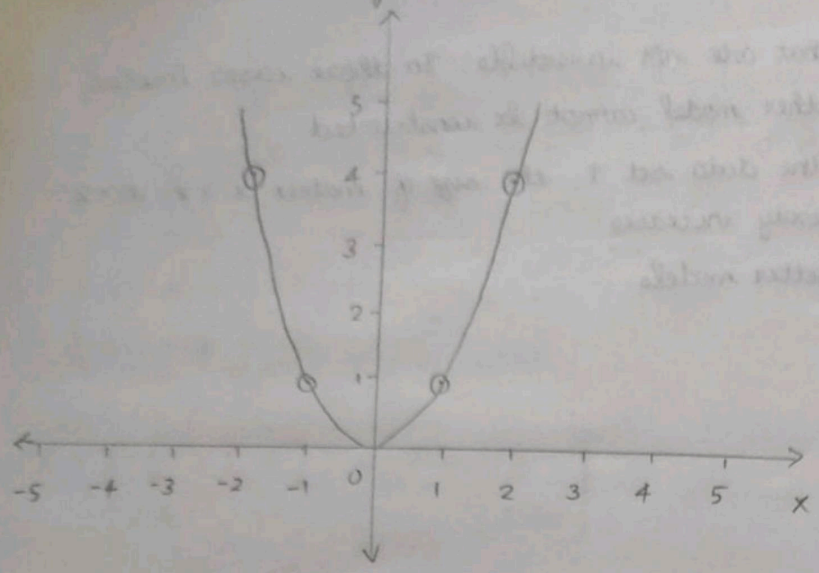


$\lambda a + (1-\lambda) b$



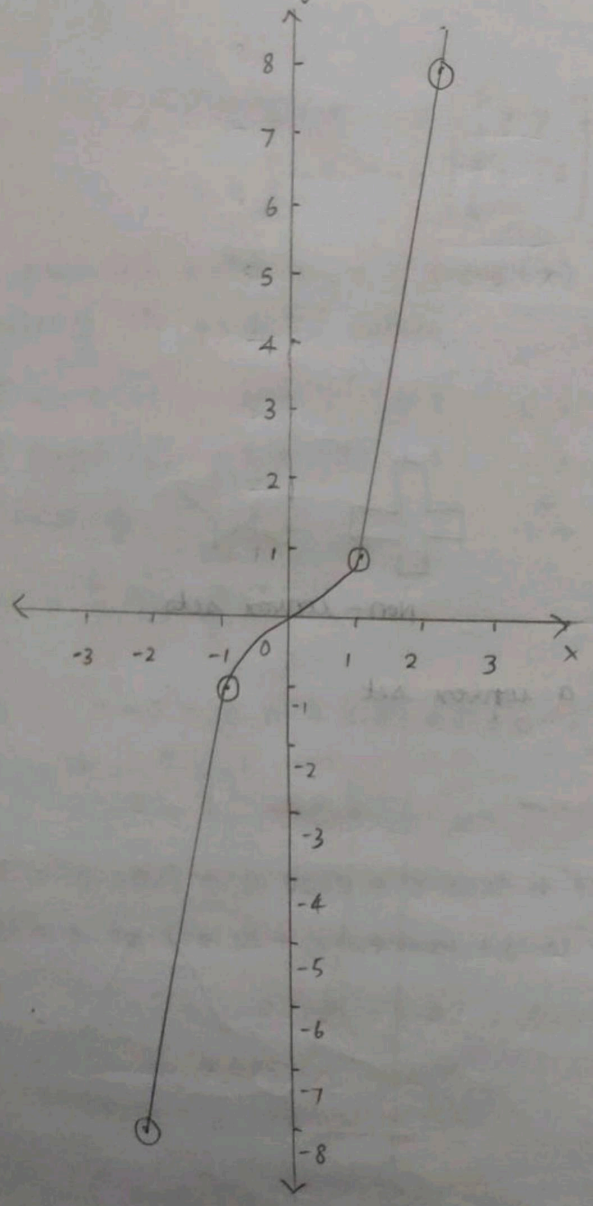
$f(\lambda a + (1-\lambda) b) \leq \lambda f(a) + (1-\lambda) f(b), \quad 0 \leq \lambda \leq 1$

$f(x) = x^2$



Convex

$f(x) = x^3$



- Negative of convex function is concave.
- Convex functions are doubly differentiable and $f''(x) \geq 0$.
- Function $f''(x) = 0$ - either convex or concave

  Eg : $y = mx + c$.

Check if the following functions are convex

i) $x \log x$ ii) $-x \log x$ iii) $e^{x \log x}$ iv) $\log x$

i) $f(x) = x \log x$

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

$f'(x) = 1 + \log x$

$f''(x) = 0 + \frac{1}{x}$

If x takes +ve values, the fn is convex.

ii) $f(x) = -x \log x$

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

$f'(x) = -1 - \log x$

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

If x takes -ve values, the fn is concave.

iii) $f(x) = e^{x \log x}$

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

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

$f(x)$ is

iv) $\log x$

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

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

$f(x)$ is not convex.

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

To check convexity,

- Find Hessian matrix H // 2nd order partial derivative matrix
- If H is positive definite / positive semi definite, f is convex.

All eigen values of H are ≥ 0 - +ve semidefinite

> 0 - +ve definite

Hessian matrix (d x d)

Jacobian

$$H = \begin{bmatrix} \partial f / \partial x \\ \partial f / \partial y \\ \partial f / \partial z \end{bmatrix}$$

$$\begin{array}{c} \\ x \\ y \\ z \end{array} \begin{array}{c} \begin{array}{ccc} x & y & z \end{array} \\ \begin{bmatrix} \frac{\partial^2 f}{\partial x^2} & \frac{\partial^2 f}{\partial x \partial y} & \frac{\partial^2 f}{\partial x \partial z} \\ \frac{\partial^2 f}{\partial y \partial x} & & \\ & & \end{bmatrix} \end{array}$$

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

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

Hessian matrix

$$\begin{array}{c} \\ x_1 \\ x_2 \\ x_3 \end{array} \begin{array}{c} \begin{array}{ccc} x_1 & x_2 & x_3 \end{array} \\ \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \frac{\partial^2 f}{\partial x_1 \partial x_3} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \frac{\partial^2 f}{\partial x_2 \partial x_3} \\ \frac{\partial^2 f}{\partial x_3 \partial x_1} & \frac{\partial^2 f}{\partial x_3 \partial x_2} & \frac{\partial^2 f}{\partial x_3^2} \end{bmatrix} \end{array}$$

Evaluating the cells of the matrix,

$\frac{\partial f}{\partial x_1} = 2(x_1 - x_2)(1)$, $\frac{\partial f}{\partial x_2} = 2(x_1 - x_2)(-1)$, $\frac{\partial f}{\partial x_3} = 4x_3$

$\frac{\partial^2 f}{\partial x_1^2} = 2$, $\frac{\partial^2 f}{\partial x_2^2} = +2$, $\frac{\partial^2 f}{\partial x_3^2} = 4$

$\frac{\partial^2 f}{\partial x_2 \partial x_1} = 2(-1)$, $\frac{\partial^2 f}{\partial x_1 \partial x_2} = -2$, $\frac{\partial^2 f}{\partial x_1 \partial x_3} = 0$

$\frac{\partial^2 f}{\partial x_3 \partial x_1} = 0$, $\frac{\partial^2 f}{\partial x_3 \partial x_2} = 0$, $\frac{\partial^2 f}{\partial x_2 \partial x_3} = 0$

The matrix, H becomes $\begin{bmatrix} +2 & -2 & 0 \\ -2 & +2 & 0 \\ 0 & 0 & +4 \end{bmatrix}$

Calculating $x^T H x$, ≥ 0

$$\underset{1\times 3}{\begin{bmatrix} x_1 & x_2 & x_3 \end{bmatrix}} \underset{3\times 3}{\begin{bmatrix} 2 & -2 & 0 \\ -2 & 2 & 0 \\ 0 & 0 & 4 \end{bmatrix}} \underset{3\times 1}{\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}}$$

$$\underset{1\times 3}{\begin{bmatrix} 2x_1 - 2x_2 & -2x_1 + 2x_2 & 4x_3 \end{bmatrix}} \underset{3\times 1}{\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}}$$

$2x_1^2 - 2x_2x_1 - 2x_1x_2 + 2x_2^2 + 4x_3^2$

$2x_1^2 - 4x_1x_2 + 2x_2^2 + 4x_3^2$

$2(x_1^2 - 2x_1x_2 + x_2^2) + 4x_3^2$

$2(x_1 - x_2)^2 + 4x_3^2 \geq 0$, +ve semi definite

$$Hx = \lambda x$$

$$Hx - \lambda Ix = 0$$

$|H - \lambda I| = 0$, H is symmetric

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

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

$$(2-\lambda)(2-\lambda)(4-\lambda) + 2(-2)(4-\lambda) = 0$$

$$(2-\lambda)^2(4-\lambda) - 4(4-\lambda) = 0$$

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

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

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

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

16 : $-4, -4$; -8

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

$(\lambda-4)(\lambda-4) = 0$

$\lambda = 4, \lambda = 4$

All eigen values ≥ 0

↳ +ve semi definite

02/08

Convex optimisation problem

min
$f(x)$ ⟶ convex function

s.t.c
$g(x) \geq 0$ ⟶ Convex set

* Unconstrained COP
* Constrained COP

Optimisation problem for L.R

Find b_0, b_1 which

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

To prove that it is convex,

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

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

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

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

$$= \sum_{i=1}^{n} 2x_i^2 \geq 0$$

$\therefore$, SSE is a convex function

Loss functions in ML:

1. SSE - LR, Binary classification
2. MSE - LR, BC
3. Negative log likelihood - Logistic regression, BC
4. Hinge loss - SVM
5. Cross entropy - Multiway classification

All these are convex functions.

Gradient descent

* Unconstrained COP
* solution based on first order partial derivative

$\min f(x)$

where $f(x)$ is a convex function.

// Real time example - Blind folded man on a hill - reach foothill - following oppo. direction of slope - as slope gives maximum ↑ - at foothill slope is 0 -

Find θ Input X, η → (step size, learning rate)

$\min f(\theta, x)$ Output θ^*

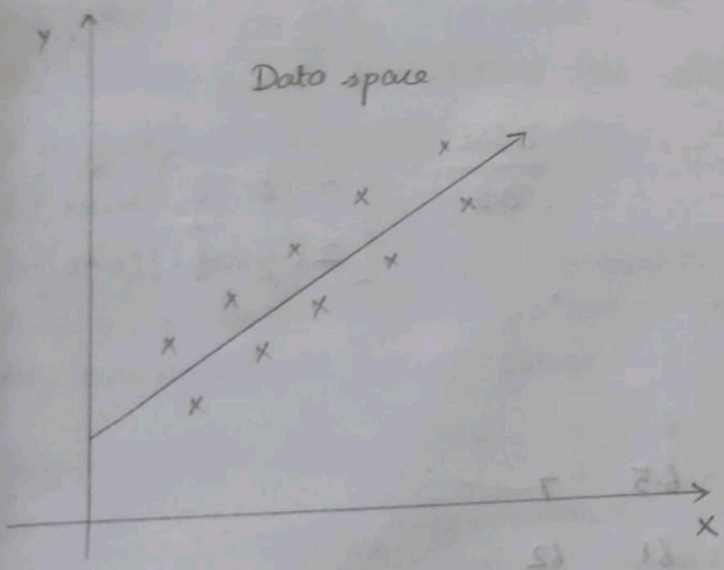
① θ = Initialize ()

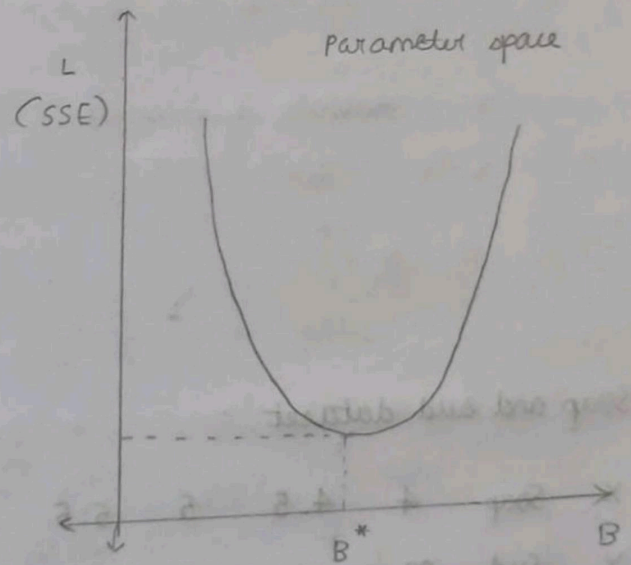
② while (! convergence)

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

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

$$\theta^* = \underset{\theta}{\operatorname{argmin}} \left(f(x, \theta) \right)$$





$B^* = \operatorname{argmin} SSE$

$$B_{new} = B_{old} - \eta \frac{\partial SSE}{\partial B}$$

// when there no change in the value of parameter for successive iterations, you can stop there.

05/08

Linear Regression using GD - Vanilla GD (Batch GD)

Input: $D = \{x_i, y_i\}_{i=1}^{n}$, η

Output: $\vec{b} = \begin{bmatrix} b_0 \\ b_1 \end{bmatrix}$

$x_i = \begin{bmatrix} 1 \\ x_i \end{bmatrix}$

1 b = initialize()

2 while(! convergence) // $b_{new} = b_{old}$ $\frac{\partial L}{\partial b} = 0$

a) for each data point (x_i, y_i)

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

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

b) $b_{new} = b_{old} - \eta \cdot grad$

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

$$\frac{\partial SSE}{\partial B} = \sum_{i=1}^{n} (y_i - \hat{y_i})(-x_i)$$

$$\frac{\partial SSE}{\partial B} = \begin{bmatrix} \frac{\partial SSE}{\partial b_0} \\ \frac{\partial SSE}{\partial b_1} \end{bmatrix}$$

$$\frac{\partial SSE}{\partial B} = \begin{bmatrix} \Sigma (y_i - \hat{y_i})(-1) \\ \Sigma (y_i - \hat{y_i})(-x_i) \end{bmatrix}$$

Soap and sud dataset

x	Soap	4	4.5	5	5.5	6	6.5	7
y	Sud	33	42	45	51	53	61	62

Assume $\eta = 0.5$

$B = \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix}$

It-1

x_0	x_1	b_0	b_1	$\hat{y}$	y	$(y_i - \hat{y_i})(-x_i)$
1	4	0.5	0.5	2.5	33	$(30.5)(-4) = -122$
1	4.5	0.5	0.5	2.75	42	$(39.25)(-4.5) = -176.625 - 122 = -298.625$
1	5	0.5	0.5	3	45	$(42)(-5) = -210 - 298.625 = -508.625$
1	5.5	0.5	0.5	3.25	51	$(47.75)(-5.5) = 262.625 - 508.625 = -771.25$
1	6	0.5	0.5	3.5	53	$(49.5)(-6) = -297 - 771.25 = -1068.25$
1	6.5	0.5	0.5	3.75	61	$(57.25)(-6.5) = -372.125 - 1068.25 = -1440.375$
1	7	0.5	0.5	4	62	$(58)(-7) = -406 - 1440.375 = -1846.375$

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

$$\beta_{new} = \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix} + \begin{bmatrix} 162.125 \\ 923.1875 \end{bmatrix} = \begin{bmatrix} 162.625 \\ 923.6875 \end{bmatrix}$$

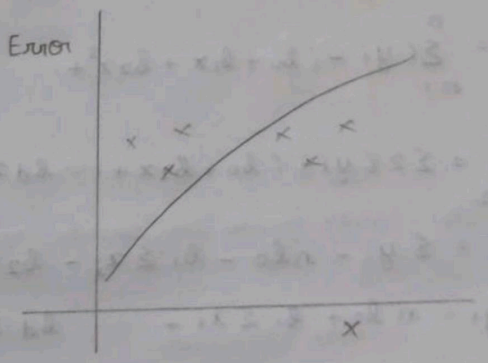
08/08

- 1epoch - calculating gradient for every DP in the training set. at the end, weights are updated.

- Vanilla GD is computationally expensive - so we go for stochastic GD.
- In stochastic GD, weights are updated w.r.t every data pt.
- Stochastic GD guarantees convergence at a faster rate; than the batch GD.
- But, the solution may be highly deviated - to overcome this we go for mini batch GD.
- In mini batch GD, the data set is split into mini batches and at the end of each mini batch, the weights are updated.

Assumptions of LR

1. All observations are independent.
2. The relationship b/w dependent and independent variable is linear.
3. Y is linear to parameters.
4. E(error) = 0
5. For a x_i, error follow normal distribution.
   For same x_i, we get different y, but we get a single $\hat{y}$, error follow $\wedge$
6. No auto correlation between errors - ind
   Generally occurs when the data is time series data.
7. Errors are homoscedasticity (constant)
   // If errors are heteroscedasticity, we go for weighted LR



- We have error in true function of LR itself

$$Y = \beta_0 + \beta_1 X + \varepsilon$$

This is because Y may depend on some other factor excluding X. ε is called the irreducible error.
On the other hand, $b_0 + b_1 x + e$, e stands for SSE which is reducible error and can be minimised.

4 - To avoid the irreducible error and to bring in all the factors Y depends on, we go for multiple regression.

- When the data points don't fit into a straight line but a curve, we go for polynomial regression.

POLYNOMIAL REGRESSION

$$Y = \beta_0 + \beta_1 x + \beta_2 x^2 + \dots \beta_d x^d + \xi$$

We have to find

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

Given,

$D = \{x_i, y_i\}$, $x_i \in \mathbb{R}$, $y_i \in \mathbb{R}$

$f: x_i \to y_i \quad \beta_0 + \beta_1 x + \beta_2 x^2 + \dots \beta_d x^d + \varepsilon$

Obj:

$\hat{f}: \mathbb{R}^d \to \mathbb{R} \quad b_0 + b_1 x + b_2 x^2 + \dots b_d x^d$

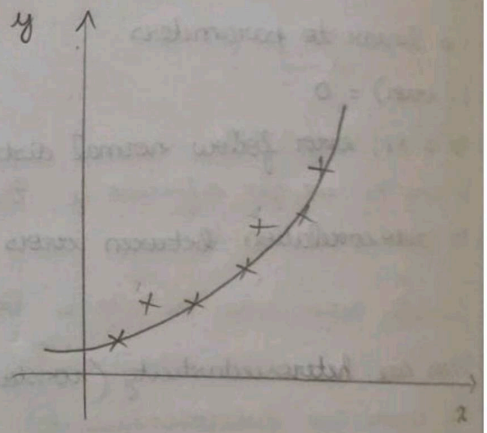
Optimisation problem - unconstrained -

Find $b = (b_0 \dots b_d)$

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

where

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



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

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

$$= \Sigma y_i - n b_0 - b_1 \Sigma x_i - b_2 \Sigma x_i^2 - \dots - b_d \Sigma x_i^d = 0$$

$$\Sigma y_i = n b_0 + b_1 \Sigma x_i + \dots b_d \Sigma x_i^d \quad \text{——} \ (1)$$

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

$$\Sigma x_i y_i = b_0 \Sigma x_i + b_1 \Sigma x_i^2 + \dots + b_d \Sigma x_i^{d+1} \quad \text{——} \ (2)$$

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

$$\Sigma x_i^2 y = b_0 \Sigma x_i^2 + b_1 \Sigma x_i^3 + \dots + b_d \Sigma x_i^{d+2} \quad \text{——} \ (3)$$

$$\frac{\partial L}{\partial \beta_d} = \Sigma 2(y_i - (\beta_0 + \beta_1 x + \dots \beta_d x^d))(-x^d)$$

$$\Sigma x_i^d y = \beta_0 \Sigma x_i^d + \beta_1 \Sigma x_i^{d+1} + \dots + \beta_d \Sigma x_i^{2d} \quad \text{---- (1)}$$

$$Y = X B$$

where

$$Y = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}_{n \times 1}, \quad X = \begin{bmatrix} 1 & x_1 & x_1^2 & \dots & x_1^d \\ 1 & x_2 & x_2^2 & & x_2^d \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_n & x_n^2 & & x_n^d \end{bmatrix}_{n \times (d+1)} \quad B = \begin{bmatrix} \beta_0 \\ \vdots \\ \beta_d \end{bmatrix}_{(d+1) \times 1}$$

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

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

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

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

Pre multiply by $(X^T X)^{-1}$ on both sides

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

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

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

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

Construct a second degree polynomial. Compare SSE with that of LR.

$$X = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 4 \\ 1 & 3 & 9 \\ 1 & 4 & 16 \\ 1 & 5 & 25 \\ 1 & 6 & 36 \end{bmatrix}_{6\times3}, \quad Y = \begin{bmatrix} 1 \\ 4 \\ 9 \\ 16 \\ 25 \\ 36 \end{bmatrix}_{6\times1}, \quad X^T = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 & 5 & 6 \\ 1 & 4 & 9 & 16 & 25 & 36 \end{bmatrix}_{3\times6}$$

$$B = (X^TX)^{-1}X^TY$$

Computing $(X^TX)^{-1}$ and X^TY

$$X^TX = \begin{bmatrix} 6 & 21 & 91 \\ 21 & 91 & 441 \\ 91 & 441 & 2275 \end{bmatrix}, \quad X^TY = \begin{bmatrix} 91 \\ 441 \\ 2275 \end{bmatrix}$$

$$(X^TX)^{-1} = \begin{bmatrix} 3.2 & -1.95 & 0.25 \\ -1.95 & 1.3696 & -0.187 \\ 0.25 & -0.187 & 0.0267 \end{bmatrix}$$

$$B = (X^TX)^{-1}X^TY$$

$$B = \begin{bmatrix} 3.2 & -1.95 & 0.25 \\ -1.95 & 1.3696 & -0.187 \\ 0.25 & -0.187 & 0.0267 \end{bmatrix} \begin{bmatrix} 91 \\ 441 \\ 2275 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

The second degree polynomial obtained is $0 + 0x + 1x^2$ i.e, x^2.

SSE = 0

Using linear regression,

$$X = \begin{bmatrix} 1 & 1 \\ 1 & 2 \\ 1 & 3 \\ 1 & 4 \\ 1 & 5 \\ 1 & 6 \end{bmatrix}_{6\times2}, \quad Y = \begin{bmatrix} 1 \\ 4 \\ 9 \\ 16 \\ 25 \\ 36 \end{bmatrix}_{6\times1}, \quad X^T = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 & 5 & 6 \end{bmatrix}_{2\times6}$$

$$X^TX = \begin{bmatrix} 6 & 21 \\ 21 & 91 \end{bmatrix}, \quad X^TY = \begin{bmatrix} 91 \\ 441 \end{bmatrix}$$

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

$$B = \begin{bmatrix} 0.8666 & -0.2 \\ -0.2 & 0.0571 \end{bmatrix} \begin{bmatrix} 91 \\ 441 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} -9.3394 \\ 6.9811 \end{bmatrix} \text{ i.e, } \begin{bmatrix} -9.333 \\ 7 \end{bmatrix}$$

The model obtained from LR is $0 + 1(x)$ i.e, x

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

x	1	2	3	4	5	6
y	1	4	9	16	25	36
$\hat{y}$	1	2	3	4	5	6

$SSE = 0 + 4 + 36 + 144 + 400 + 900 = 1484$

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

The model is $-9.333 + 7x$

x	1	2	3	4	5	6
y	1	4	9	16	25	36
$\hat{y}$	-2.333	4.667	11.667	18.667	25.667	32.667

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

$$= 11.108889 + 0.444889 + 7.112889 + 7.112889 + 0.444889 + 11.10889$$

$SSE = 37.333334$

$SSE_{LR} > SSE_{PR}$

10)8

Apply polynomial regression on Iris dataset between every pair of features and construct an optimal model (which has lowest SSE / maximum r^2) based on the evaluation metrics