ML ⇒ Field of study that gives the computers the ability to learn without being explicitly programmed.

Date:

Linear Regression

Agenda behind learning any algorithm :

1. What problem we are solving.
2. Geometric intuition
3. Mathematical intuition.

Data come with input x and output label y.

ML

- Supervised
  - Regression (continuous outcome)
  - Classification
- Unsupervised

Data comes with only input x, but not output label y. Algorithm has to find structure in the data.

Linear Regression ⇒

Training Dataset
↓
Train the model
↓
Create Hypothesis
↓
Predictions

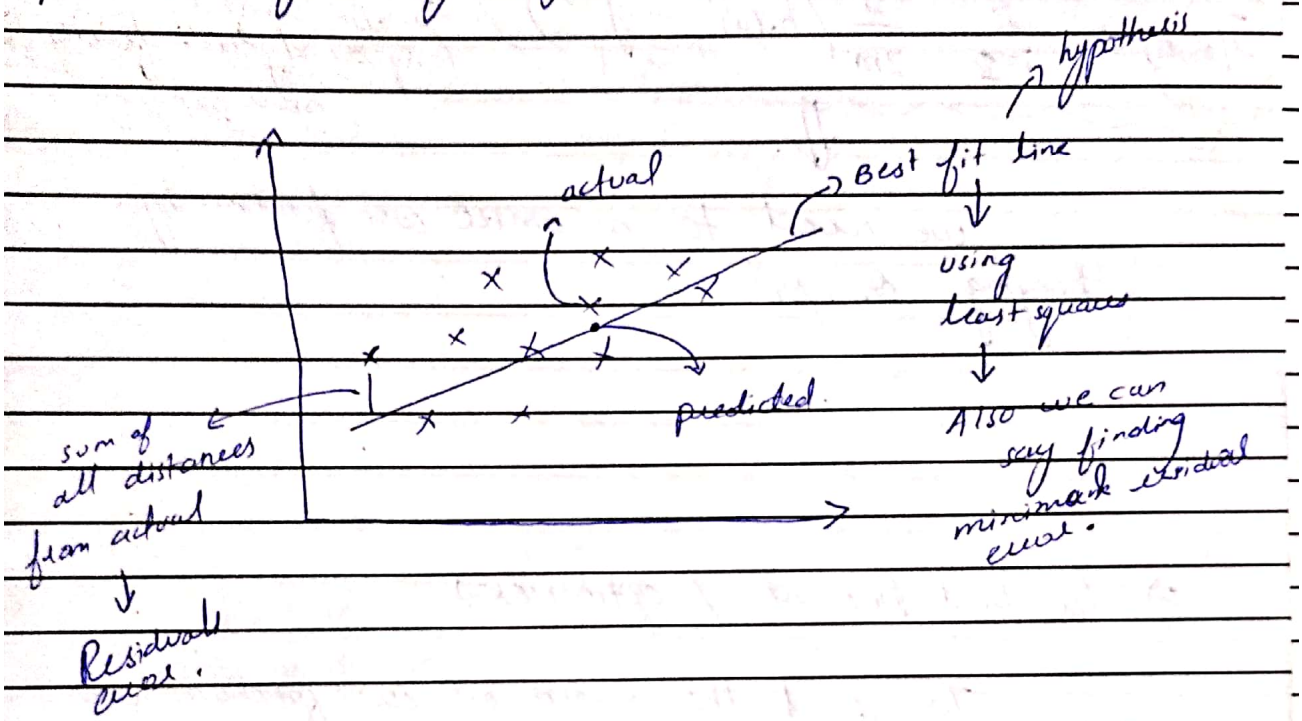
LUCKY WRITE SERIES

Scanned with CamScanner

⇒ The more opportunities you give to the learning algorithm, the better it would perform.

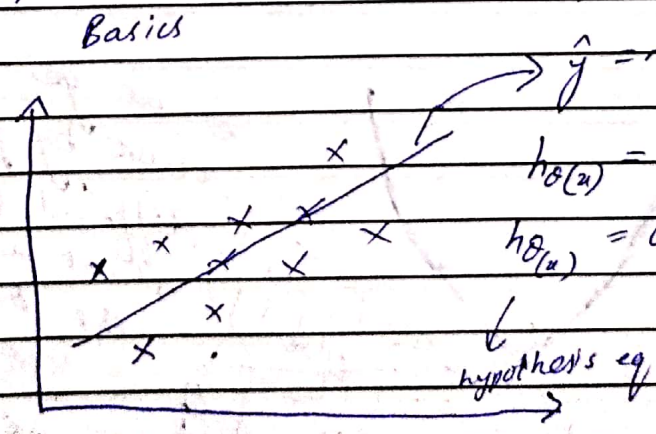
Date:

For example: we are given data of height & weight of a person, and we want to create model which will predict height using weight.



Maths ⇒

Basics



$\hat{y} = mx + c$

$h_{\theta(x)} = \theta_0 + \theta_1 x_1$

$h_{\theta(x)} = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 + \cdots + \theta_n x_n$.

hypothesis eq.

We need to find best θ's (giving with minimal error)

Date:

⇒ Cost function;

$$J(\theta_0, \theta_1) \Rightarrow \sum_{i=1}^{m} \frac{1}{2m} \left(\overbrace{h_\theta(x)^{(i)}}^{\text{predicted}} - \overbrace{y^{(i)}}^{\text{actual}} \right)^2$$

m = no. of data points

⇒ also called Squared Error function

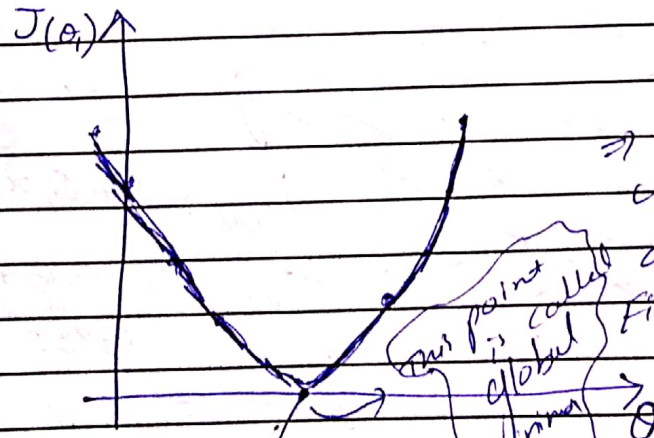
(least squares)

⇓

we need to minimize cost function by changing θ_0, θ_1.

⇒ Gradient Descent / optimizer ⇒

To find the minimized cost function with minimal error.



This is the best θ_1, as it give minimal cost function.

This point is called global minima

⇒ we get the curve by plotting different θ_1. First, we will use the θ_1, to create line (hypo), then we will use gradient descent to find best hypothesis.

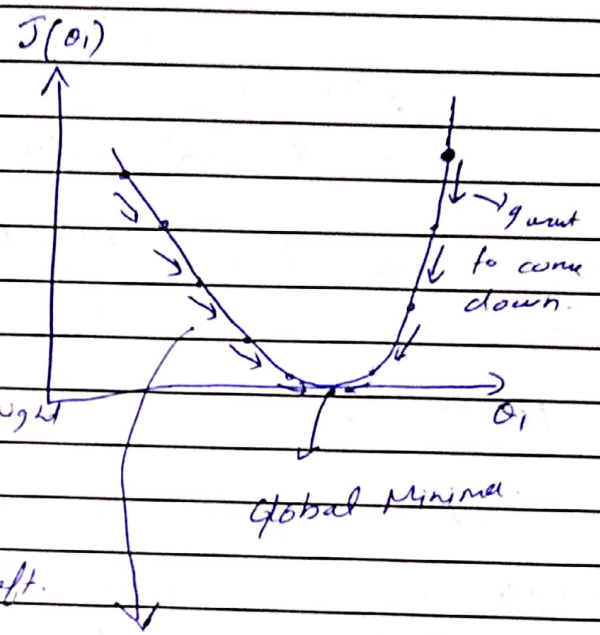
first we initialize θ_1, then repeat converge theorem, (decrement or increment) θ_1 to find (optimal solution) ↓ where slope is nearest to zero (or global minima)

Date:

⇒ Repeat Convergence Theorem ⇒
(Gradient Descent)

$$\theta_j = \theta_j - \alpha \frac{\partial(J(\theta_1))}{\partial \theta_j}$$

learning rate → slope



→ slope on right

$\theta_1 := \theta_1 - \alpha\,(+ve)$

$\theta_1 := \theta_1 - \alpha\,(-ve)$

↳ slope on left.

∴ learning rate decides speed (of movement)

$$Var(mean) = \frac{SS_{(mean)}}{n} \qquad Var(fit) = \frac{SS(fit)}{n}$$

$$R^2 = \frac{Var(mean) - Var(fit)}{Var(mean)}$$

→ R^2 tells how much variation in dependent feature can be explained by taking independent features into account.

→ Suppose if $R^2 = 100\%$, then we can say independent feature explains 100% variation in dependent feature.

LUCKY WRITE SERIES

Scanned with CamScanner

Date:

⇒ We need a way to determine if the R^2 value is statistically significant.

↳ For this, we need to calculate a p-value.

This equation will tell us if R^2 is significant.

$$F = \frac{SS(mean) - SS(fit) / (p_{fit} - p_{mean})}{SS(fit) / (n - p_{fit})}$$

degrees of freedom.

For example;

As example of weight → height give in 2 pages before, we can say;

$$R^2 = \frac{\text{Variation in height explained by weight}}{\text{Variation in height without taking weight into acc}}$$

$$F = \frac{\text{Variation in height explained by weight}}{\text{Variation in height not explained by weight}}$$

n = number of data points.

p_{fit} = number of parameter in fitted line.

p_{mean} = number of parameters in mean line.

e.g.:- $y = mx + c$ — 2 parameters

$y = mx_1 + mx_2 + c$ — 3 parameters.

LUCKY WRITE SERIES

Scanned with CamScanner

★ Supervised Learning ⇒ you give your learning algorithm, the examples to learn from which includes the right answers. By seeing correct pairs of input and our desired output y, the learning algorithm train itself to just take the input results output

Date:

By Andrew NG

More on ML Fundamentals : Linear Regression

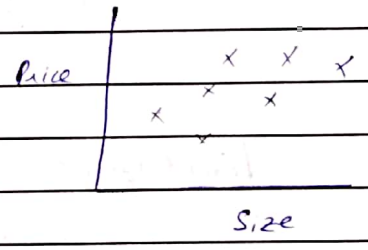
Autonomous car → Supervised Learning Algorithm Problem → Regression.

Indepth intuition using house price predictor problem;

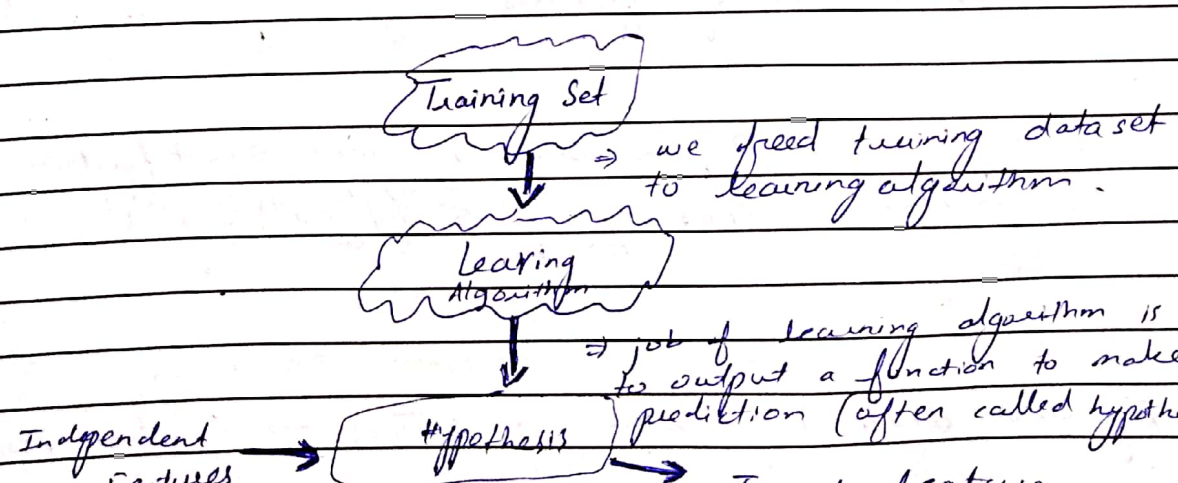
House Size	No. of bed rooms	House price
....	. .	. .
. .	. .	. .
. .	. .	. .

↓ Independent Features

↓ Target Feature



In supervised learning problem;



Scanned with CamScanner

Input (X)	→	output (Y)	Application
email	→	spam? (0/1)	spam filtering
audio	→	text transcripts	speech recognition
english	→	spanish	machine translation
Ad, user info	→	click? (0/1)	online advertising
image, radar info	→	position of other cars	self driving car

Date:

How to represent h?

$$h_{\theta_0,\theta_1}(x) = h(x) = \theta_0 + \theta_1 x$$ ← linear regression model

OR

$$h_{\theta_0,\theta_1,\theta_2}(x) = h(x) = \theta_0 + \theta_1 \underset{\text{size}}{x_1} + \theta_2 \underset{\text{No. of bed room}}{x_2}$$ ⇒ features

General Form;

$$h(x) = \sum_{j=0}^{n} \theta_j x_j$$

where $x_0 = 1$

(x^i, y^i)
⇓
- represent first
- data point in training set.

TERMINOLOGY:

θ → parameters, coefficient, weights
m → No. of data points (or training examples)
x → features (or inputs)
y → target feature / target variable / output
(x, y) → Data point / training example (single training examples)
n → No. of features

→ Main goal → choose θ's → which minimize → cost function.

$$\text{Cost Function} = J(\theta) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$$

→ Algorithm which helps minimizing → $J(\theta)$ → Gradient Descent

Iterative Algorithm

LUCKY WRITE SERIES

Scanned with CamScanner

classification problem ⇒ small number / finite set of possible outputs.
unsupervised learning ⇒ Find something interesting (like structures and patterns) in unlabeled data.

Date:

⇒ Gradient Descent; ⇒ steepest descent ⇒?

1) → Start with some θ's (say $\theta = \vec{0}$).
2) → Keep changing θ's to reduce $J(\theta)$.
3) → simultaneous update of the parameters θ's.

assignment operator in usual mathematics. (not a equality)

learning rate

$$\theta_j := \theta_j - \alpha \left\{ \frac{\partial}{\partial \theta_j} J(\theta) \right\}$$

→ for intuition you can say remember that derivative in which direction you are taking baby step in combination with α, it also determines the size of the step.

→ It basically controls how big a step you take downhill.

$$\Rightarrow \frac{\partial}{\partial \theta_j} J(\theta) = \frac{\partial}{\partial \theta_j} \frac{1}{2m} (h_\theta(x) - y)^2$$

$$= 2\left(\frac{1}{2m}\right)(h_\theta(x) - y) - \frac{\partial}{\partial \theta_j}(h_\theta(x) - y)$$

$$= \frac{1}{m}(h_\theta(x) - y) \frac{\partial}{\partial \theta_j}(\theta_0 x_0 + \theta_1 x_1 + \theta_n x_n - y)$$

$$= \frac{1}{m}(h_\theta(x) - y) \cdot x_j$$

eg. For θ_1 and θ_0 as a parameter

$$\theta_0 := \theta_0 - \alpha \frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1)$$

$$\theta_1 := \theta_1 - \alpha \frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1)$$

So our equation becomes;

#

↓ on next page

$$\theta_j := \theta_j - \alpha \sum_{i=1}^{m} \frac{(h_\theta(x)^{(i)} - y^{(i)})}{m} x_j^{(i)}$$

∴ where $x_j = x_0 = 1$

Repeat until convergence → meaning repeat until you come near to global minima.

Here ⇒ i ⇒ $1 - m$ (Total number of data points) (to)
⇒ j ⇒ $0 - (n-1)$ (No. of features)

LUCKY WRITE SERIES

Scanned with CamScanner

Unsupervised learning → clustering algorithm. → group similar data points together.
→ Anomaly Detection → Find unusual data points.
→ Dimensionality reduction → compress data using fewer numbers.

Date:

Gradient Descent → which uses summation of all datapoints (in $J(\theta)$) → in every iteration

⇒ ~~continued.~~ temp ⇒ temporary

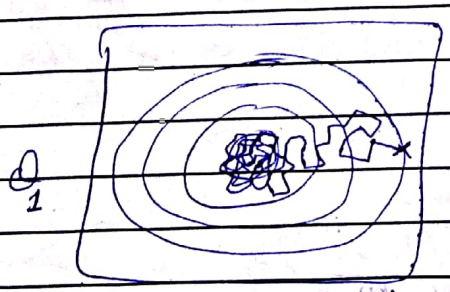
correct: simultaneous update	Incorrect
$temp_\theta_0 := \theta_0 - \alpha \frac{\partial J(\theta_0,\theta_1)}{\partial \theta_0}$	$temp_\theta_0 := \theta_0 - \alpha \frac{\partial J(\theta_0,\theta_1)}{\partial \theta_0}$
$temp_\theta_1 := \theta_1 - \alpha \frac{\partial J(\theta_0,\theta_1)}{\partial \theta_1}$	$\theta_0 := temp_\theta_0$
$\theta_0 := temp_\theta_0$	$temp_\theta_1 := \theta_1 - \alpha \frac{\partial J(\theta_0,\theta_1)}{\partial \theta_1}$
$\theta_1 := temp_\theta_1$	$\theta_1 := temp_\theta_1$

Batch Gradient Descent.
↓
slow (Due to huge calculation every iteration)
↓
due to reading / scanning over all values.

Alternative to batch gradient descent → Stochastic Gradient Descent.

Stochastic Gradient Descent → which uses only one datapoint → in every iteration
↓
fast
↓
don't give accurate result
↓
generate noise.

e.g.



⇒ oscillates near global minima on average.

↳ probably make fine prediction even if you don't converge at global minima.
↳ commonly used for large data sets.

LUCKY WRITING

Scanned with CamScanner

or with number of iteration

When to stop stochastic gradient descent → plot the $J_{(\theta)}$ over time.

once it stop going down.

Date: stop iteration

Linear regression has no local minima / local optimum.

It only has global minimum. due to parabolic function.

→ Vectorization increases speed due to parallel computing.

⇒ Normal Equation → Shortcut Method to jump to global minima in one step → but can only be used for linear regression.

Normal Equation Derivation;

$$\nabla_\theta \cdot J_{(\theta)} = \begin{bmatrix} \partial J/\partial \theta_0 \\ \partial J/\partial \theta_1 \\ \partial J/\partial \theta_2 \end{bmatrix}$$

$\theta \in \mathbb{R}^{n+1}$

using LA

$A \in \mathbb{R}^{2\times 2} \Rightarrow A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}$

If $f(A) = A_{11} + A_{12}^2$ given that $f(\mathbb{R}^{2\times 2}) \to \mathbb{R}$

$\Rightarrow f\left(\begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix}\right) = 5 + 6^2$

$$\nabla_A f(A) = \begin{bmatrix} \partial f(A)/\partial A_{11} & \partial f(A)/\partial A_{12} \\ \partial f(A)/\partial A_{21} & \partial f(A)/\partial A_{22} \end{bmatrix} = \begin{bmatrix} 1 & 2A_{12} \\ 0 & 0 \end{bmatrix}$$

Main Goal ⇒ $\nabla_\theta J(\theta) = \vec{0}$.

Some Impact ant Properties

① ⇒ $\operatorname{tr} A = \operatorname{tr} A^T$

② ⇒ If $f(A) = \operatorname{tr} AB \Rightarrow \nabla_A f(A) = B^T$

③ ⇒ $\operatorname{tr} AB = \operatorname{tr} BA$ & $\operatorname{tr} ABC = \operatorname{tr} CAB$

④ ⇒ $\nabla_A \operatorname{tr} AA^TC = CA + C^TA$

LUCKY WRITE SERIES

Scanned with CamScanner

Date:

$$J(\theta) = \frac{1}{2} \sum_{i=1}^{m} \left(h(x^{(i)}) - y^{(i)} \right)^2$$

If $X = \begin{bmatrix} \text{---} (x^{(1)})^T \text{---} \\ \text{---} (x^{(2)})^T \text{---} \\ \text{---} (x^{(3)})^T \text{---} \\ \vdots \\ \text{---} (x^{(m)})^T \text{---} \end{bmatrix}$

Design Matrix

$$X\theta = \begin{bmatrix} \text{---} (x^{(1)})^T \text{---} \\ \text{---} (x^{(2)})^T \text{---} \\ \vdots \\ \text{---} (x^{(m)})^T \text{---} \end{bmatrix} \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \end{bmatrix} = \begin{bmatrix} x^{(1)T}\theta \\ x^{(2)T}\theta \\ x^{(3)T}\theta \\ \vdots \\ x^{(m)T}\theta \end{bmatrix}$$

$$\begin{bmatrix} h_\theta(x^{(1)}) \\ \vdots \\ h_\theta(x^{(m)}) \end{bmatrix}$$

$$y = \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(m)} \end{bmatrix}$$

As $\Rightarrow J(\theta) = \frac{1}{2} \sum_{i=1}^{m} \left(h(x^{(i)}) - y^{(i)} \right)^2 = \frac{1}{2} (X\theta - y)^T (X\theta - y)$

$\therefore z^T z =$

$$\Rightarrow \nabla_\theta J(\theta) = \nabla_\theta \frac{1}{2} (X\theta - y)^T (X\theta - y) = 0$$

$$= \frac{1}{2} \nabla_\theta (X^T\theta^T - y^T)(X\theta - y) = 0$$

$$= \frac{1}{2} \nabla_\theta (X^T\theta^T X\theta - X^T\theta^T y - X\theta y^T + y^T y) = 0$$

$$= \frac{1}{2} [X^T X\theta + X^T X\theta - X^T y - X^T y] = 0$$

$$= X^T X\theta - X^T y = 0$$

$$\Rightarrow X^T X\theta = X^T y$$

$$\boxed{\theta = (X^T X)^{-1} X^T y}$$

LUCKY

Scanned with CamScanner

Date:

ML Fundamentals :- ~~Logistic Regression~~

→ Locally Weighted Regression.

choice b/w them depends upon the data and b/w the desired trade-off b/w simplicity and complexity.

Parametric Learning Algorithm → fit fixed set of parameter (θ_i) to data → Linear Regression

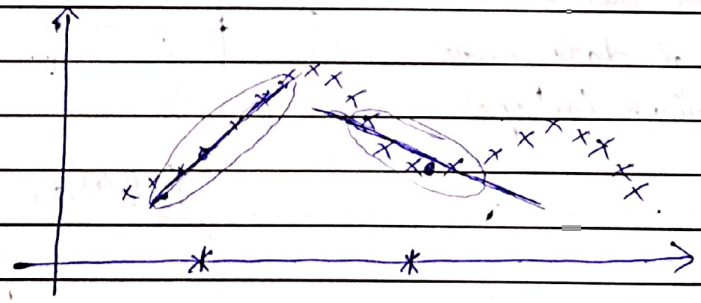
make strong assumptions about data's functional form with a fixed number of parameters, offering efficient training but limited flexibility.

Non-Parametric Learning Algorithm → Amount of data/parameters you need to keep grows (linearly) with size of data.

Non-parametric learning, like KNN or DT's etc., adapt their complexity based on the data, making them more flexible but potentially computationally expensive with large datasets.

Graphical Intuition ⇒

Here $J(\theta) = \sum_{i=1}^{m} w^{(i)} (y^{(i)} - \theta^T x^{(i)})^2$



Fit θ to minimize cost function which is $\sum_{i=1}^{m} w^{(i)} (y^{(i)} - \theta^T x^{(i)})^2$

where $w^{(i)}$ is a weight function.

$$w^{(i)} = \exp\left(-\frac{(x^{(i)} - x)^2}{2\tau^2}\right)$$

∴ If $|x^{(i)} - x|$ is small, $w^{(i)} \approx 1$

∴ If $|x^{(i)} - x|$ is large $w^{(i)} \approx 0$

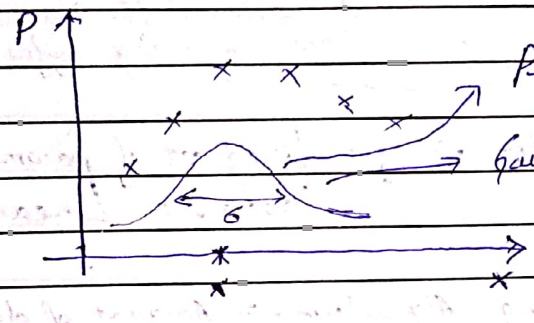
LUCKY WHITE SERIES

Scanned with CamScanner

→ locally weighted Regression is good for dataset with less features and huge data.

Date:

⇒ Intuition about gaussian distribution ⇒



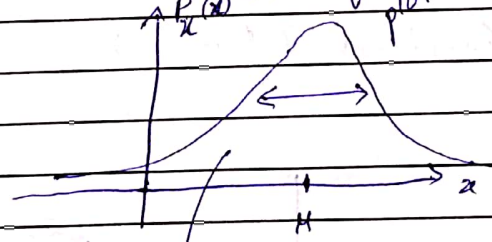
Probability density function.

Gaussian Bell Curve.

$$P = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{(x-\mu)^2}{2\sigma^2}\right]$$

⇓ Probability density function of random variable e^{x} plotted against x.

$P_x(x)$ ⇒



Here, standard deviation would mean the average deviation (or distance) of data points from the mean (or input value).

$$\boxed{\int_{-\infty}^{\infty} P_x(x)\,dx = 1} \Rightarrow \frac{\text{number}}{\text{total}}$$

What is standard deviation?

A standard deviation is how much data scatter around the mean value.

$$\sigma = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \mu)^2}$$

Standard Deviation is a measure of how much the data is dispersed around the mean. low standard deviation means data are clustered around the mean, and high standard deviation indicates data are more spread c

LUCKY WRITERS

Scanned with CamScanner

Standard deviation & variance ⇒ What's the difference?

- The standard deviation is the average distance from the mean.
- The variance is the squared average distance from the mean.

Date:

⇒ Intuition about kernel density estimation ⇒

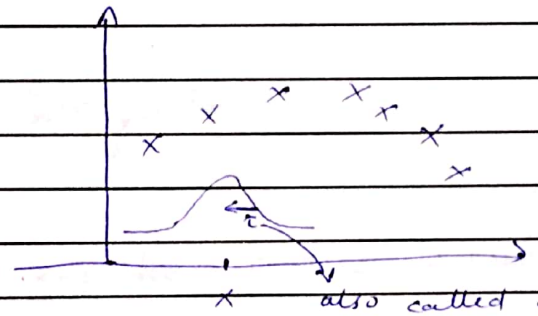
This is weight function.

Hs $w^{(i)} = \exp\left(\frac{(x^{(i)} - x)^2}{2\tau^2}\right)$

Different from gaussian density function

i.e. → doesn't integrate to 1.

where τ (tau) is the band with parameter.



also called bandwith parameter

It controls the width of the kernel function used to estimate density.

used in kernel density estimation.

⇒ bandwith parameter and standard deviation are both used in different context and have different meanings.

→ Bandwith parameter → kernel density estimation (which is non-parametric method for estimating the probability density function of a random variable).

→ Standard deviation → describes the spread of a distribution.

LUCKY WHITE SERIES

Scanned with CamScanner