MACHINE LEARNING

Experiment Name :	Expt. No.
	Date
Problem Statement :	
	Week No.

Definition : A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if the performance on task T as measured by P increases with experience E.

→ Supervised learning & unsupervised learning

Algorithm is fed the "right answers". Task of the algo is to predict more right ans

i) Regression - predict continuous real valued o/p.

ii) classification - Discrete value o/p

Algo is not fed with labels / instructions on performing a task.
Used to organise clusters of data & identify them.
eg - Social n/w analysis
Market segmentation
Astronomical data analysis
The algo is responsible for finding structure in the data

* REGRESSION

Notations

m = number of training examples

x = input variable / features

y = output variable / target variable. to be predicted.

1 training example - (x, y)

ith " " $(x^{(i)}, y^{(i)})$

Supervised learning algorithm → Training example

Training set $(x^{(i)}, y^{(i)})$ $i = 1 \dots m$
↳ Training set

↓

Learing algorithm

↓

i/p (x) → hypothesis 'h' → o/p (estimated value) (y)

$h : X \to Y$

$h_\theta(x) = \theta_0 + \theta_1 x$ → Linear regression with one variable
univariate linear regression.

When the target variable that is to ~~pre~~ be predicted is continuous, it is called a regression problem.

When y takes a small no. of discrete values → classification problem.

* COST FUNCTION

θ_i's → parameters

With diff θ_i's we get different hypothesis.

Scanned by CamScanner

Time series classification

→ Model based : Auto regression & HMM

→ distance based : Dynamic time warping

→ Feature based : extract meaningful features like DFT, STFT, DWT, PCA, SVD etc

Automatic feature based approaches using deep learning models → convolutional neural networks.

Training all parameters is done jointly using back propogation method.

Problem Statement :

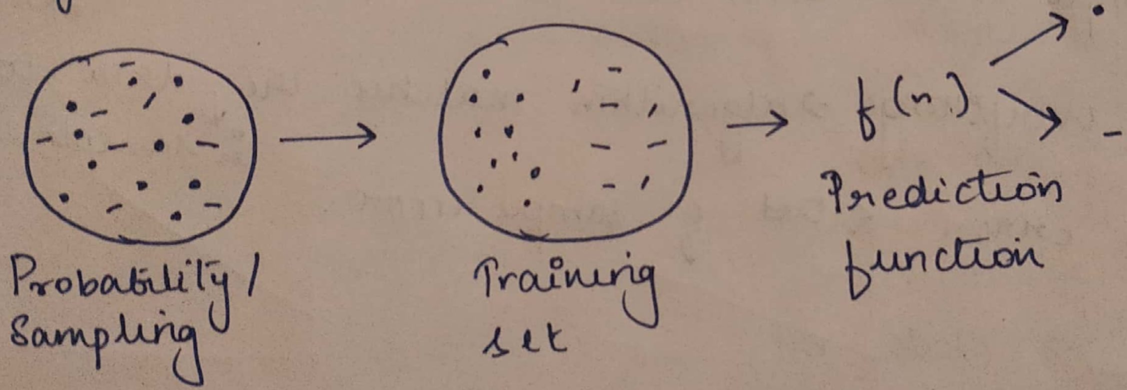
Date

Week No.

Central dogma of prediction

Any set of data requires probability/sampling to create a training set that differentiates ~~do~~ the various samples.

We then extract characteristics from the data set by which we can model a function on to predict on new samples as to which class it belongs to.



Components of a predictor

question → i/p data → features → algorithm → parameters → evaluation.

(what to Predict ?)

(characters)

(prediction function definition)

Properties of good features

→ Lead to data compression - We need to collect signi features of lesser size.

→ Retains relevant info

→ Are created based on expert application knowledge.

Prediction is about accuracy tradeoffs

→ Interpretability vs accuracy

→ Speed vs accuracy

→ Simplicity vs accuracy

→ Scalability vs accuracy.

* In sample error - the error rate you get on the same data that was used to build the predictor. RE~~DISTRI~~SUBSTITUITION ERROR

* Out of sample error - error rate on new data set. GENERALIZ ERROR.

* Reason for overfitting → algorithm matches the data too well in sample.

* In sample error < Out of sample error.

Problem Statement :

Date

Week No.

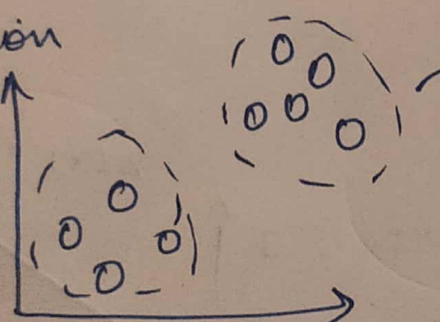
Supervised Learning - Right answers are given prior to building the model.

Regression - Prediction of continuous valued o/p.

Classification - Discrete valued output (0 or 1) (0,1,2..)

Unsupervised learning - no information on the data is given. The task is to find some structure to the data which can later on, be used to extract some information

Eg :-



The algo might decide based on the data ~~the~~ it could be separated as shown - Clustering algo.

We can derive structure from the data where we dont necessarily know the effect of the variables.
We can obtain this structure by clustering data based on relationships among the variables in the data.

* No feedback based on the prediction results.

Scanned by CamScanner

Model Representation

$x^{(i)} \rightarrow$ i/p (features) $y^{(i)} \rightarrow$ o/p (target) $m \rightarrow$ number of features

$i = 1 \dots m \rightarrow$ training set.

$(x^{(i)}, y^{(i)}) \rightarrow$ training example.

Flow of supervised learning :

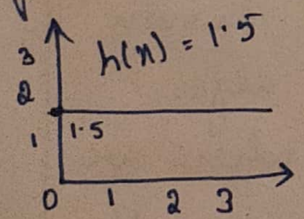
Training set
↓
Learning algo
↓
$x \rightarrow \boxed{h} \rightarrow$ predicted y
hypothesis

Goal is : given a training set, to learn a fn 'h': $x \rightarrow y$ so that $h(x)$ is a good predictor for the corresponding value of y.

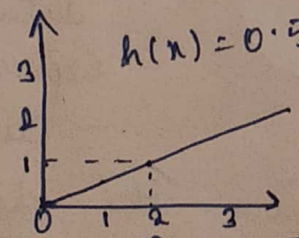
$h_\theta(x) = \theta_0 + \theta_1(x) \rightarrow$ Univariate Linear regression

θ_i's $\rightarrow$ parameter of the model.

Eg: -

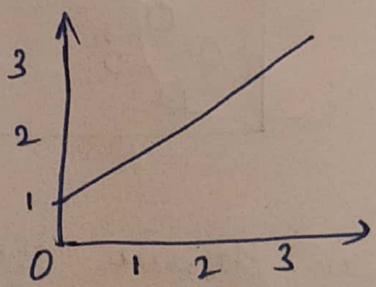


$\theta_0 = 1.5 \quad \theta_1 = 0$
$h(x) = 1.5$



$\theta_0 = 0 \quad \theta_1 = 0.5$
$h(x) = 0.5x$

$h(x) = 1 + 0.5x$



$\theta_0 = 1 \quad \theta_1 = 0.5$
$h(x) = 1 + 0.5x$

Idea : Choose θ_0, θ_1 such that $h(x)$ is close to y for our training example (x, y)

$$\therefore \underset{\theta_0 \theta_1}{\text{minimize}} \ \frac{1}{2M} \sum_{i=1}^{M} \left(h_\theta(x^{(i)}) - y^{(i)} \right)^2$$

// why are we averaging? $\frac{1}{2M}$?

Experiment Name : Square error func or Squared Mean Error

Problem Statement :

Expt. No.

Date

Week No.

cost function → $J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2$

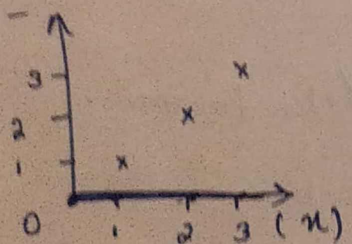
$\underset{\theta_0\ \theta_1}{\text{minimize}}$

$\underset{\theta_0\ \theta_1}{\text{minimize}}$ $\underbrace{J(\theta_0, \theta_1)}_{\text{Cost func.}}$

Basically the difference between the predicted value and the actual value.

// $\frac{1}{2}$ used as convenience for gradient descent because it can later be used to cancel out when derivative is taken.

Eg :-



$h_\theta(x) = \theta_1 x$ // simplified version

$J(\theta_1) = \frac{1}{2m} \sum_{i=1}^{m} (h(x) - y)^2$

what is $J(0) = ?$

$h_\theta(x^i) = ?$ for every value of x, $h_\theta(x) = 0$

$J(\theta_1) = \frac{1}{2 \times 3} \sum_{i=1}^{3} [(0-1)^2 + (0-2)^2 + (0-3)^2]$

$= \frac{1}{6} [1 + 4 + 9] = \frac{14}{6}$

YAY!.

*Gradient descent - Algorithm used for minimizing any fn.

Repeat until Convergence {

$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1)$ for $j = 0$ & $j = 1$

$:=$ → Assigment

α → learning rate

α ↑↑ v. large → Aggressive gradient descent

&

We want to $\min_{\theta_0,\theta_1} J(\theta_0, \theta_1)$ by gradient descent

① start with an arbituary θ_0, θ_1

② keep varrying θ_0 & θ_1 until minimum is acheived.

* Simultaneously update θ_0 & θ_1

step 1 : $temp0 := \theta_0 = \theta_0 - \alpha \frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1)$

$temp1 := \theta_1 = \theta_1 - \alpha \frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1)$

step 2 : $\theta_0 = temp0$

$\theta_1 = temp1$

Eg :- $\theta_0 = 1$ $\theta_1 = 2$, find the update.

$tempo :- \theta_0 = 1 - \alpha \frac{\partial}{\partial \theta_0} J(\overset{1}{\theta_0}, \overset{2}{\theta_1})$

where $J(\theta_0, \theta_1) = \frac{1}{2M} \sum_{i=1}^{M} (h_\theta(x^i) - y^i)^2$? // Not finished.

// Vector = $n \times 1$ matrix. n depicts the dimension.

Can add & sub matrices only if they are of the same dimensions.

Prediction = Data matrix × Parameters.

Scanned by CamScanner

Learning Problems in ML :-

Classification - assign a category.

Regression - Predict a real value

Rank - order items : in

clustering - Seperation of region into homogeneous segme

Dimensionality reduction /Manifold learning - learn a lower dimensional representation.

→ Supervised learning : Classification , Regression

→ Un " " : Density estimation, clustering, dimensionality reduction.

→ Semi - supervised "

→ Reinforcement " etc.

Decision Trees

Basic algorithm

1. Start with all variables in one group
2. Find the variable / split that best seperates the outcomes
3. Divide data into two groups ("leaves") on that split ("node")
4. With in each split find the best variable / sp that seperates the outcomes.
5. Continue till the groups are too small / sucintly

Measures of Impurity.

$$\hat{P}_{mk} = \frac{1}{N_m} \sum_{x_i \text{ in leaf } m} 1(y_i = k)$$

→ signifies the number of times that particular class appears in that leaf

↓

Probability to estimate

m^{th} leaf

k^{th} class

N → no. of objects in that class.

Misclassification Error → $1 - \hat{P}_{mk(m)}$; $k(m)$ = most common class.

0 = perfect purity

0.5 = no purity. → when leaves are perfectly balanced; we don't get homogeneity.

If 1, then perfect purity in another direction?

Gini Index:-

$$\sum_{k \neq k'} \hat{P}_{mk} \times \hat{P}_{mk'} = \sum_{k=1}^{K} \hat{P}_{mk}(1 - \hat{P}_{mk}) = 1 - \sum_{k=1}^{K} P_{mk}^2$$

→ Same for this.

Deviance / info gain

$$- \sum_{k=1}^{K} \hat{P}_{mk} \log_2 \hat{P}_{mk}$$

1 = no purity 0 = purity.