

Module 2: Understanding data - 2.

2.1. Bivariate data & multivariate data

2.2. multivariate statistics

2.3. Essential mathematics for multivariate data.

2.4. Feature engineering & dimensionality reduction techniques

2.5. Basic learning theory: Design of learning system.

2.6. Introduction to concept of learning

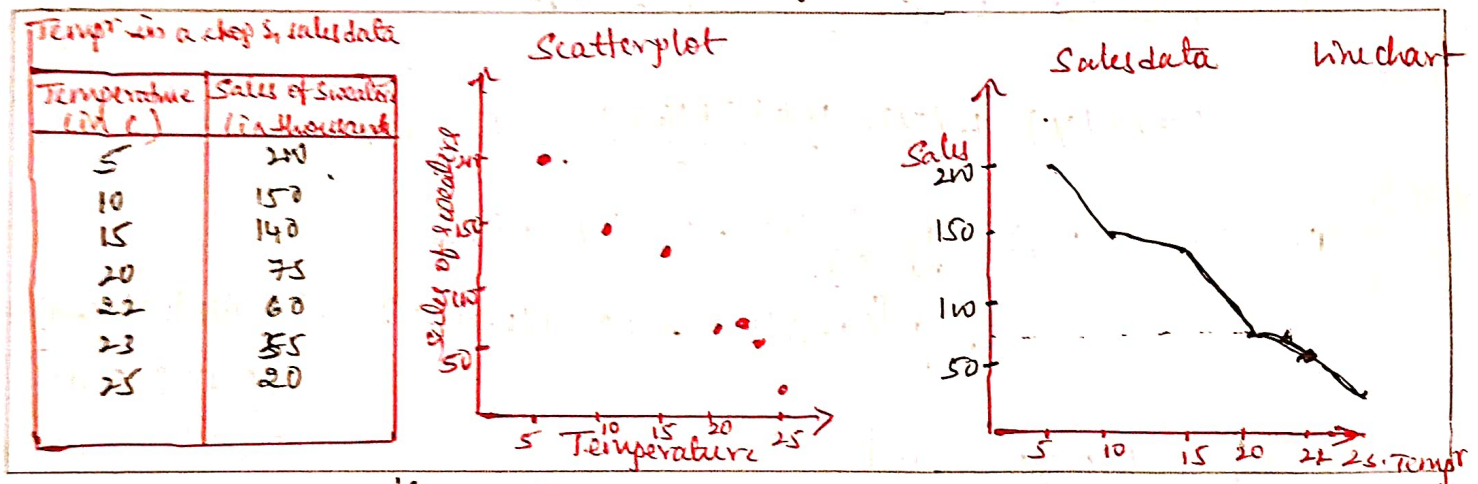
2.7. Modeling in machine learning

2.8.

2.1. Bivariate data & multivariate data.

Bivariate data involves 2 variables.

Aim is to find relationship b/n among data.



The relationships can be used in comparisons, finding causes & in further exploration. To do that, graphical display of the data is necessary. One such graph method is called Scatter plot. & It is used to visualize bivariate data.

Scatter plot is useful to plot 2 variables with & without nominal variables, to illustrate the trends & also to show differences.

It is a plot b/n explanatory & response variables. It is a 2D graph showing the relationship b/n 2 variables.

The Scatter plot i.e. indicates strength, shape, direction & the presence of outliers

Line graphs are similar to scatter plots.

Bivariate Statistics:-

Covariance & correlation are example of Bivariate statistics. Covariance is a measure of joint probability of random variables X & Y . It is defined as Covariance (X, Y) or $COV(X, Y)$ & is used to measure the variance b/n 2 dimensions.

$$COV(X, Y) = \frac{1}{N} \sum_{i=1}^N (x_i - E(X)) (y_i - E(Y))$$

x_i & y_i are data values from X & Y . $E(X)$ & $E(Y)$ are the means values of x_i & y_i .

N is the no of given data.

$COV(X, Y)$ is same as $COV(Y, X)$.

Find the covariance of data $X = \{1, 2, 3, 4, 5\}$ & $Y = \{1, 4, 9, 16, 25\}$

Solⁿ: Mean(X) = $E(X) = \frac{1+2+3+4+5}{5} = \frac{15}{5} = 3$

Mean(Y) = $E(Y) = \frac{1+4+9+16+25}{5} = \frac{55}{5} = 11$.

$$\begin{aligned} \text{Covariance}(X, Y) &= \frac{1}{N} \sum_{i=1}^N (x_i - E(X)) (y_i - E(Y)) \\ &= \frac{1}{5} [(1-3)(1-11) + (2-3)(4-11) + (3-3)(9-11) + (4-3)(16-11) + (5-3)(25-11)] \end{aligned}$$

$COV(X, Y) = 12$

Correlation:

It measures the strength & direction of a linear relationship b/n the x & y variables.

The correlation indicates the relationship b/n dimensions using its sign.

The sign is more important than the actual value.

- The sign is more important than the actual value.
- (1) If the value is +ve, it indicates that the dimensions increase together.
 - (2) If the value is -ve, it indicates that while one - dimension increases, the other dimension decreases.

- (3) If the value is zero, it indicates that both the dimensions are
- If the dimensions are correlated, then it is better to remove one dimension as it is a redundant dimension.

(2)

If the given attributes are $X = (x_1, x_2, \dots, x_N)$ & $Y = (y_1, y_2, \dots, y_N)$.
 then the Pearson correlation coefficient, 'r' is given as

$$r = \frac{\text{Cov}(X, Y)}{\sigma_X \sigma_Y}$$

where σ_X, σ_Y are the standard deviations of X & Y .

$$\sigma_X = \sqrt{\frac{\sum (x_i - \bar{x})^2}{N}} \quad \text{where } \bar{x} = \text{mean} = \frac{\sum x_i}{N}$$

$$\sigma_X = 1.41$$

$$\sigma_Y = 8.646$$

$$r = \frac{\text{Cov}(X, Y)}{\sigma_X \sigma_Y} = \frac{12}{1.41 \times 8.646} = 0.984$$

Multivariate statistics:

In ML, almost all datasets are multivariable.

Multivariate data is the analysis of more than 2 observable variables.
 & often, thousands of multiple measurements need to be conducted for one/more subjects.

Multivariate data may have more than 2 dependent variables

Some of the multivariate analysis are

* Regression analysis

* Principal component Analysis

* Path analysis.

Id	Attribute1	Attribute2	Attribute3
1	1	5	9
2	2	6	1
3	3	7	2
4	4	8	3

The mean of multivariate data is a mean vector & the mean of the above 3 attributes is given as $(2.5, 6.5, 3.75)$.

$$\frac{26.5}{4} = 6.5$$

The variance of multivariate data becomes the covariance matrix.

$$\frac{15}{4} = 3.75$$

The mean vector is called centroid & variance is called dispersion matrix.

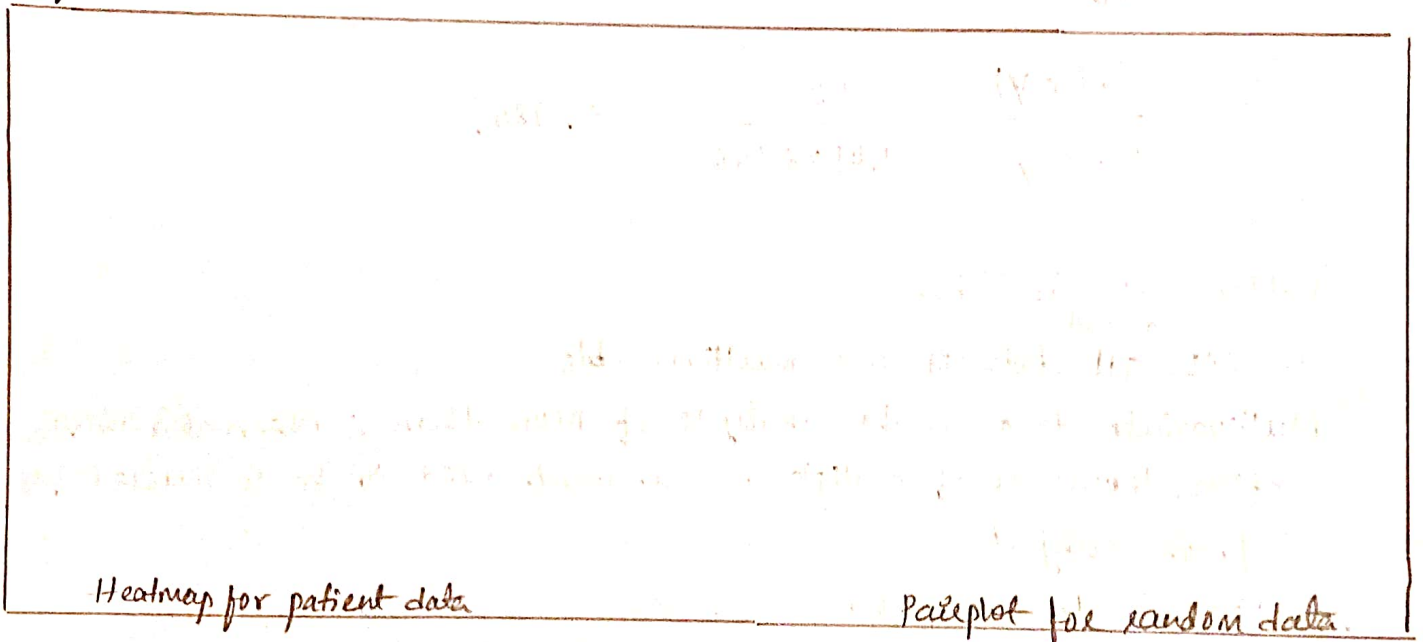
Heat map:

It is a graphical representation of 2D matrix. It takes a matrix as input & colours it.

The darker colours indicate very larger values, & lighter colours indicate smaller values.

Advantage: Humans perceive colours well. So, by colour shading, larger values can be perceived well.

e.g: In vehicle traffic data, heavy traffic regions can be differentiated from low traffic regions through heatmap.



Pairplot / Scatter matrix is a data visual technique for multivariate data. A Scatter matrix consists of several pair-wise scatterplots of variables of the multivariate data.

2.3. Essential Mathematics for multivariate data.

ML involves many mathematical concepts from the domain of linear algebra, statistics, probability & information theory.

2.3.1. Linear systems & gaussian elimination for multivariate data.

A linear system of equations is a group of equations with unknown variables

Let $Ax = y$, then the solution x is given by

$$x = \frac{y}{A} = A^{-1}y.$$

This is true if y is not zero & A is not zero.

The logic can be extended for N -set of eqns with ' n ' unknown variables,

i.e. if $A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{pmatrix}$ & $y = (y_1, y_2, \dots, y_n)$; then the

unknown variable x can be computed as $x = y/A = A^{-1}y$.

If there is a unique solution, then the system is called consistent independent.

If there are various solutions, then the system is called consistent dependent.

If there are no solutions & if the eqns are contradictory, then the system is called inconsistent.

For solving larger no of system of eqns, gaussian elimination can be used.

The procedure for applying gaussian elimination is given as follows

- (1). Write the given matrix
 - (2). Append vector y to the matrix A . This matrix is called augmentation matrix
 - (3). Keep the element a_{11} as pivot & eliminate all a_{i1} in second row using the matrix operation.
 $R_2 - \frac{a_{21}}{a_{11}}$, here R_2 is the 2nd row & $\frac{a_{21}}{a_{11}}$ is called the multiplier.
- The same logic can be used to remove a_{i1} in all other eqns.

4. Repeat the same logic & reduce it to reduced echelon form
Then, the unknown variable as

$$x_n = \frac{y_n}{a_{nn}}$$

5. Then, the remaining unknown variables can be found by back substitution as

$$x_{n-1} = \frac{y_{n-1} - a_{n-1,n} x_n}{a_{(n-1)(n-1)}} \quad \text{This part is called backward substitution}$$

To facilitate the application of Gaussian elimination method, the following row operations are applied.

- (1) Swapping the rows
- (2) Multiplying/dividing a row by a constant
- (3) Replacing a row by adding/subtracting a multiple of another row to it.

Solve the following set of equations using Gaussian elimination method.

$$2x_1 + 4x_2 = 6$$

$$4x_1 + 3x_2 = 7$$

Soln: Rewrite the eqn in matrix form $\left(\begin{array}{cc|c} 2 & 4 & 6 \\ 4 & 3 & 7 \end{array} \right)$

Apply the transformation by dividing the row 1 by 2.

There are no general guidelines of row operations other than reducing the given matrix to row echelon form.

$$\left(\begin{array}{cc|c} 1 & 2 & 3 \\ 4 & 3 & 7 \end{array} \right)$$

$$R_2 = R_2 - 4R_1$$

$$\left(\begin{array}{cc|c} 1 & 2 & 3 \\ 0 & -5 & -5 \end{array} \right)$$

$$R_2 = R_2 / 5$$

$$= \left(\begin{array}{cc|c} 1 & 2 & 3 \\ 0 & 1 & 1 \end{array} \right)$$

\therefore in the reduced echelon form, it can be observed that $x_2 = 1$, $x_1 = 1$

2.3.2. Matrix decompositions:-

It is often necessary to reduce a matrix to its constituent parts so that complex matrix operations can be performed. These methods are known as matrix factorization methods / eigen decomposition.

It is the way to reducing the matrix into eigen values & eigen vectors.

Then, the matrix 'A' can be decomposed as

$$A = Q \Lambda Q^T$$

Where Q is the matrix of eigen vectors.

Λ is the diagonal matrix

Q^T = Transpose matrix of Q

LU decomposition:-

It is the simplest matrix decomposition. where matrix A ^{can be} decomposed into matrices.

$$A = LU$$

Where L = Lower triangular matrix

U = Upper triangular matrix..

The decomposition can be done using Gaussian elimination method.

Find LU decomposition of the given matrix $A = \begin{pmatrix} 1 & 2 & 4 \\ 3 & 3 & 2 \\ 3 & 4 & 2 \end{pmatrix}$

Solⁿ: First, augment an Identity matrix & apply Gaussian elimination

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 2 & 4 \\ 3 & 3 & 2 \\ 3 & 4 & 2 \end{bmatrix} \text{ Initial matrix}$$

$$\begin{bmatrix} 1 & 0 & 0 \\ 3 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 2 & 4 \\ 0 & -3 & -10 \\ 3 & 4 & 2 \end{bmatrix} \quad R_2 = R_2 - 3R_1$$

$$\begin{bmatrix} 1 & 0 & 0 \\ 3 & 1 & 0 \\ 3 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 2 & 4 \\ 0 & -3 & -10 \\ 0 & -2 & -10 \end{bmatrix} \quad R_3 = R_3 - 3R_1$$

$$\begin{bmatrix} 1 & 0 & 0 \\ 3 & 1 & 0 \\ 3 & \frac{2}{3} & 1 \end{bmatrix} \begin{bmatrix} 1 & 2 & 4 \\ 0 & -3 & -10 \\ 0 & 0 & -\frac{10}{3} \end{bmatrix} \quad R_3 = R_3 - \frac{2}{3}R_2$$

$$\therefore L = \begin{pmatrix} 1 & 0 & 0 \\ 3 & 1 & 0 \\ 3 & \frac{2}{3} & 1 \end{pmatrix}$$

$$U = \begin{pmatrix} 1 & 2 & 4 \\ 0 & -3 & -10 \\ 0 & 0 & -\frac{10}{3} \end{pmatrix}$$

If the matrix is large, LU decomposition method ^{can be} used.

2.4. Feature engineering & dimensionality reduction Techniques.

Features are attributes. Feature engineering is about determining the subset of features that form an important part of the I/P that improves the performance of the model.

Feature engineering deals with 2 problems

- * Feature transformation - Extraction of features & creating new features that may be helpful in increasing performance.
- * Feature Selection.

Feature subset selection is another important aspect of feature engineering. that focuses on selection of features to reduce the time but not at the cost of reliability.

The subset selection reduces the dataset size by removing irrelevant features & constructs a minimum set of attributes for ML.

Typically, the feature subset selection problem uses greedy approach. by looking for the best choice at the time using locally optimal choice while hoping that it would lead to global optimal solⁿ.

The features can be removed based on 2 aspects.

- (1) Feature relevancy: Some features contribute more for classification than other features.
e.g.: A mole on the face can help in face detection than common features like nose. i.e. Features should be relevant.

The relevance of the features can be determined based on information measures such as mutual information, correlation based features like correlation coefficient & distance measures.

- (2) Feature redundancy: Some features are redundant.
e.g. when a database table has a field called DOB, then age field is not relevant as age can be computed easily from DOB.
This helps in removing the column age that leads to reduction of dimension.

The procedure / Summary is

- (1) Generate all possible subsets.
- (2) Evaluate the subsets & model performance
- (3) Evaluate the results for optimal feature selection.

Stepwise forward selection:-

This procedure starts with an empty set

Every time, an attribute is tested for statistical significance for best quality & is added to the reduced set. This process is continued till a good reduced set of attributes is obtained.

Stepwise backward Elimination

This procedure starts with a complete set of attributes. At every stage, the procedure removes the worst attribute from the set, leading to the reduced set

Combined approach: Both forward & reverse methods can be combined so that the procedure can add the best attribute & remove the worst attribute.

Principal Component Analysis (PCA):

The idea of the PCA is to transform a given set of measurements to a new set of features so that the features exhibit high information packing properties. This leads to a reduced & compact set of features.

Basically, this elimination is made possible because of the information redundancies. This compact representation is of a reduced dimension.

Consider a group of random vectors of the form

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

The mean vector of the set of random vector is defined as $m_x = E[x]$

The operator 'E' refers to the expected value of the population.

This is calculated theoretically using the Probability density function (PDF) of the elements x_i & the joint probability density functions b/w the elements x_i & x_j

From this, the covariance matrix can be calculated as

$$C = E[(x - m_x)(x - m_x)^T]$$

For M random vectors, when M is large enough, the mean vector & covariance matrix can be approximately calculated as

$$m_x = \frac{1}{M} \sum_{k=1}^M x_k$$

$$A = \frac{1}{M} \sum_{k=1}^M x_k x_k^T - m_x m_x^T$$

This covariance matrix is real & symmetric.

If e_i & λ_i be the set of eigen vectors & corresponding eigen values of the covariance matrix, the eigen values can be arranged in a descending order so that $\lambda_1 > \lambda_2 > \lambda_{i+1}$ for $i=1, 2, \dots, n-1$

The corresponding eigen vectors are calculated

Based on this, the transform kernel is constructed.

Let the transform kernel be 'A', then, the matrix rows are formed from the eigen vectors of the covariance matrix.

The mapping of the vectors x to y using the transformation can now be described as

$$y = A(x - m_x) \text{ - This is called as Hotelling transform.}$$

The original vector x can now be reconstructed as follows

$$x = A^T y + m_x.$$

The goal of PCA is to reduce the set of attributes to a newer, smaller set that captures the variance of the data.

The variance is captured by fewer components, which would give the same result as the original, with all the attributes.

If 'K' largest eigen values are used, the recovered information would be

$$x = A_K^T y + m_x.$$

The advantages of PCA are immense. It reduces the attribute list by eliminating all irrelevant attributes.

The PCA algorithm is as follows

1. The target dataset x is obtained.
2. The mean is subtracted from the dataset. $(x - m) \Rightarrow$ Transform the dataset with zero mean.
3. The covariance of dataset x is obtained. Let it be 'C'
4. Eigen values & eigen vectors of the covariance matrix are calculated
5. The eigen vector of the highest eigen value is the Principal component of the dataset. The eigen values are arranged in descending order. The feature vector is formed with these eigen vectors in its columns
Feature vectors = {eigen vector₁, eigen vector₂, ... - eigen vector_n}
6. Obtain the Transpose of feature vector. Let it be 'A'

7. PCA transform is $y = A^T(x - m)$, where x is the P/P dataset, m is the mean, A is the transpose of the feature vector.

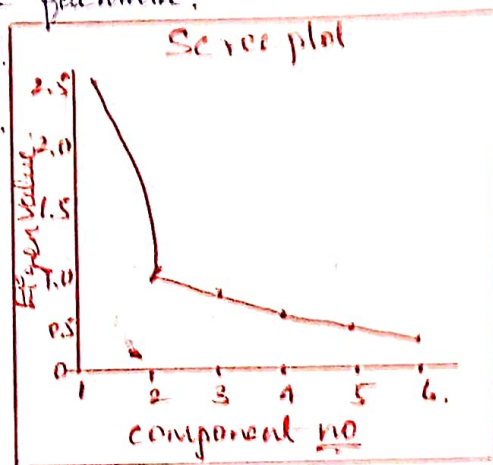
The original data can be retrieved using the formula.

$$\begin{aligned}\text{Original data (f)} &= \{A^{-1}xy\} + m \\ &= \{A^T xy\} + m.\end{aligned}$$

The new data is a dimensionally reduced matrix that represents the original data.

\therefore PCA is effective in removing the attributes that do not contribute.

If the original data is required, it can be obtained with no loss of information.



Scree plot is a visualization technique to visualize the principal components/variables that play a more important role as compared to other attributes.

From the above scree plot, one can infer the relevance of the attributes.

The scree plot indicates that the first attribute is more important than all other attributes.

Apply PCA & find the transformed data for the datapoints $\begin{pmatrix} 2 \\ 6 \end{pmatrix}$ & $\begin{pmatrix} 1 \\ 7 \end{pmatrix}$.

& again, apply the inverse & prove that PCA works

Solⁿ: One can combine 2 vectors into a matrix as follows.

The mean vector can be computed $m_x = \frac{1}{M} \sum_{k=1}^M x_k$

$$M = \begin{pmatrix} \frac{2+1}{2} \\ \frac{6+7}{2} \end{pmatrix} = \begin{pmatrix} 1.5 \\ 6.5 \end{pmatrix}$$

As part of PCA, the mean must be subtracted from the data to get the adjusted data.

$$x_1 = \begin{pmatrix} 2 - 1.5 \\ 6 - 6.5 \end{pmatrix} = \begin{pmatrix} 0.5 \\ -0.5 \end{pmatrix}$$

$$x_2 = \begin{pmatrix} 1 - 1.5 \\ 7 - 6.5 \end{pmatrix} = \begin{pmatrix} -0.5 \\ 0.5 \end{pmatrix}$$

One can find the covariance for these data vectors.

The covariance can be obtained using $A = \frac{1}{M} \sum_{k=1}^M x_k x_k^T = m_x m_x^T$

$$m_1 = \begin{pmatrix} 0.5 \\ -0.5 \end{pmatrix} \begin{pmatrix} 0.5 & -0.5 \end{pmatrix} = \begin{pmatrix} 0.25 & -0.25 \\ -0.25 & 0.25 \end{pmatrix}$$

$$m_2 = \begin{pmatrix} -0.5 \\ 0.5 \end{pmatrix} (-0.5, 0.5) = \begin{pmatrix} 0.25 & -0.25 \\ -0.25 & 0.25 \end{pmatrix}$$

The final covariance matrix is obtained by adding the two matrices as
 $A = \begin{pmatrix} 2 & 2 \\ 1 & 3 \end{pmatrix} \rightarrow (x^T - \bar{x})^T (x - \bar{x}) = \begin{pmatrix} x & y \\ 0 & x \end{pmatrix} = \begin{pmatrix} 2 & 2 \\ 1 & 3 \end{pmatrix}$
 $C = \begin{pmatrix} 0.5 & -0.5 \\ -0.5 & 0.5 \end{pmatrix}$

The eigen values & eigen vectors of matrix C can be obtained as $\lambda_1 = 1, \lambda_2 = 0$
 The eigen vectors are $\begin{pmatrix} -1 \\ 1 \end{pmatrix}$ & $\begin{pmatrix} 1 \\ 1 \end{pmatrix}$.

The matrix 'A' matrix can be obtained by packing the eigen vector of these eigen values of matrix 'C'
 For this problem, $A = \begin{pmatrix} -1 & 1 \\ 1 & 1 \end{pmatrix}$

The transpose of A, $A^T = \begin{pmatrix} -1 & 1 \\ 1 & 1 \end{pmatrix}$ is also the same matrix as it is.
 an orthogonal matrix.

The matrix can be normalized by dividing each element of the vector, by the norm of the vector to get

$$A = \begin{pmatrix} -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{pmatrix}$$

$$\begin{pmatrix} (1-\lambda) & 2 \\ 5 & 4-\lambda \end{pmatrix} = 0$$

$$(1-\lambda)(4-\lambda) - 10 = 0$$

$$4 - \lambda - 4\lambda + \lambda^2 - 10 = 0$$

One can check that the PCA matrix 'A' is orthogonal. $\lambda^2 - 5\lambda - 6 = 0$

'A' matrix is orthogonal. $A^T = A$ & $A \cdot A^T = I$

$$\lambda = 6, \lambda = -1$$

$$A A^T = \begin{pmatrix} -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{pmatrix} \begin{pmatrix} -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

when $\lambda = 6, (A - \lambda I) x = 0$

$$\begin{vmatrix} 1-6 & 2 \\ 5 & 4-6 \end{vmatrix} \begin{bmatrix} -5 & 2 \\ 5 & -2 \end{bmatrix} \begin{bmatrix} -5 & 2 \\ 5 & -2 \end{bmatrix}$$

$$\begin{bmatrix} -5 & 2 \\ 5 & -2 \end{bmatrix} \begin{bmatrix} 9 \\ 5 \end{bmatrix} =$$

$$y = A^T y + m x$$

$$y = A(x - m) = \begin{pmatrix} -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{pmatrix} \begin{pmatrix} 0.5 & -0.5 \\ -0.5 & 0.5 \end{pmatrix} \begin{bmatrix} -5 & 2 \\ 5 & -2 \end{bmatrix}$$

$$= \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ 0 & 0 \end{pmatrix}$$

$$\begin{bmatrix} 9 \\ 5 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

One can check the original matrix can be retrieved from the matrix as

$$x = \{A^T x y\} + m$$

$$x = A^T y + m = \begin{pmatrix} -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{pmatrix} \begin{pmatrix} -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ 0 & 0 \end{pmatrix} + \begin{pmatrix} 1.5 \\ 6.5 \end{pmatrix}$$

$$= \begin{pmatrix} \frac{1}{2} & -\frac{1}{2} \\ -\frac{1}{2} & \frac{1}{2} \end{pmatrix} + \begin{pmatrix} 1.5 \\ 6.5 \end{pmatrix} = \begin{pmatrix} 2 & 1 \\ 6 & 7 \end{pmatrix}$$

Linear Discriminant Analysis.

It is also a feature reduction technique like PCA.

The focus of LDA is to project higher dimension data to a line

LDA is used to classify the data.

Let there be 2 classes, C_1 & C_2 .

Let μ_1 & μ_2 be the mean of the patterns of 2 classes.

The mean of the class C_1 & C_2 can be computed as

$$\mu_1 = \frac{1}{N_1} \sum_{x_i \in C_1} x_i \quad \mu_2 = \frac{1}{N_2} \sum_{x_i \in C_2} x_i$$

The aim of LDA is to optimize the function $J(V) = \frac{V^T \sigma_B V}{V^T \sigma_W V}$.

Where, V is the linear projection & σ_B & σ_W are class scatter matrix & within scatter matrix, respectively.

For the 2-class problem, these matrices are given as.

$$\sigma_B = N_1 (\mu_1 - \mu) (\mu_1 - \mu)^T + N_2 (\mu_2 - \mu) (\mu_2 - \mu)^T$$

$$\sigma_W = \sum_{x_i \in C_1} (x_i - \mu_1) (x_i - \mu_1)^T + \sum_{x_i \in C_2} (x_i - \mu_2) (x_i - \mu_2)^T$$

The maximization of $J(V)$ should satisfy the eqⁿ

$$\sigma_B V = \lambda \sigma_W V \quad (\text{or}) \quad \sigma_W^{-1} \sigma_B V = \lambda V.$$

As $\sigma_B V$ is always in the direction of $(\mu_1 - \mu_2)$, V can be given as

$$V = \sigma_W^{-1} (\mu_1 - \mu_2)$$

Let $V = (v_1, v_2, \dots, v_d)$ be the generalized eigen vectors of σ_B & σ_W . where 'd' is the largest eigen values as in PCA.

The transformation of x is given as $y = V_d^T x$.

Singular Value Decomposition.

It is another useful decomposition technique. It is useful in compression.
Let 'A' be the matrix, then the matrix A can be decomposed as

$$A = U S V^T.$$

where A is the given matrix of dimension $m \times n$, U is the orthogonal matrix whose dimension is $m \times m$.

S is the diagonal matrix of dimension $n \times n$.

V is the orthogonal matrix.

The procedure for finding decomposition matrix is given as follows

1. For a given matrix, find $A A^T$.
2. Find eigen values of $A A^T$.
3. Sort the eigen values in a descending order. Pack the eigen vectors as a matrix U.
4. Arrange the square root of the eigen values in diagonal. This matrix is diagonal matrix, S.
5. Find eigen values & eigen vectors for $A^T A$. Find the eigen value & pack the eigen vector as a matrix called V.

Thus, $A = U S V^T$; where U & V are orthogonal matrices.

The columns of U & V are left & right singular values respectively.

~~SVD is~~

$$a_{ij} = \sum_{k=1}^n U_{ik} S_k V_{jk}.$$

Based on the choice of retention, the compression can be controlled.

Find the SVD of the matrix $A = \begin{pmatrix} 1 & 2 \\ 4 & 9 \end{pmatrix}$

$$(1). A \cdot A^T = \begin{pmatrix} 1 & 2 \\ 4 & 9 \end{pmatrix} \begin{pmatrix} 1 & 4 \\ 2 & 9 \end{pmatrix} = \begin{pmatrix} 5 & 22 \\ 22 & 97 \end{pmatrix}$$

The eigen value & eigen vector of this matrix can be calculated to get U.

The eigen values of this matrix are 0.0098 & 101.9902.

The eigen vectors of the matrix are $U_1 = \begin{pmatrix} 0.2268 \\ 1 \end{pmatrix}$ $U_2 = \begin{pmatrix} -4.4086 \\ 1 \end{pmatrix}$

These vectors are normalized to get the vectors respectively as

$$U_1 = \begin{pmatrix} 0.2212 \\ 0.9752 \end{pmatrix}, U_2 = \begin{pmatrix} -0.9752 \\ 0.2212 \end{pmatrix}.$$

The 'U' matrix can be obtained by concatenating the above vectors U_1 & U_2 as

$$U = (U_1, U_2) = \begin{pmatrix} 0.2212 & -0.9752 \\ 0.9752 & 0.2212 \end{pmatrix}$$

The matrix 'V' can be obtained by finding $A^T A$.

~~It is~~ $V = \begin{pmatrix} 17 & 38 \\ 38 & 85 \end{pmatrix}$

The eigen values are 0.0098 & 101.9902

The eigen vectors can be found as follows.

$$V_1 = \begin{pmatrix} 0.447 \\ 1 \end{pmatrix} \text{ when } \lambda = 101.99$$

$$V_2 = \begin{pmatrix} -2.236 \\ 1 \end{pmatrix} \text{ when } \lambda = 0.0098$$

The above V_1 & V_2 can be normalized as follows

$$V_1 = \begin{pmatrix} 0.4082 \\ 0.9129 \end{pmatrix}$$

$$V_2 = \begin{pmatrix} -0.9129 \\ 0.4082 \end{pmatrix}$$

The matrix 'V' can be obtained by concatenating the above vectors

$$\text{as } V = [V_1, V_2] = \begin{pmatrix} 0.4082 & -0.9129 \\ 0.9129 & 0.4082 \end{pmatrix}$$

The matrix 'S' can be found as the diagonal matrix as

$$S = \begin{pmatrix} \sqrt{101.9902} & 0 \\ 0 & \sqrt{0.0098} \end{pmatrix} = \begin{pmatrix} 10.099 & 0 \\ 0 & 0.099 \end{pmatrix}$$

\therefore , the matrix decomposition $A = U S V^T$ is complete.

Advantage of SVD is to reduce the contents of image while retaining the quality of the image. It is useful in data reduction

- Problems!**
- Find mean, median, mode, standard deviation & variance for a given univariate dataset $S = [5, 10, 15, 20, 25, 30]$ of marks.
 - Find arithmetic mean & geometric mean for a given univariate dataset $S = [5, 10, 15, 20, 25, 30]$ of marks.
 - Find 5 point summary & plot the boxchart for a given univariate dataset $S = [5, 10, 15, 20, 25, 30]$ of marks.
 - Perform the descriptive analysis of data for the below table (a) & (b).

Table (a): Sample data

Age	Weight
1	4.2
2	4.5
3	4.7
4	5.2
5	6
6	6.2
7	7
8	7.2
9	7.5
10	8.5

Table (b): Students marks table.

Standard	English	Hindi	Maths	Science
1	45	70.5	90	40
2	60	72.5	80	45
3	60	80	90	50
4	80	80	90	80
5	85	72	70	60

- Find min & max marks scored in each subject
 - Find details of student who scored highest marks in maths
 - Find the students with marks English > 60 & Maths > 70 .
5. For univariate attribute such as weight, English & math marks, find the following
- Mean, median, mode.
 - Weighted mean, Geometric mean & Harmonic mean
 - Variance & standard deviation
 - Coefficient of variance
 - Skewness & kurtosis.
 - 5-point summary, IQR, Semi-quantile
6. For the Bivariate data such as English & Math, find.
- Covariance & correlation b/n the variables
 - Covariance b/n English & Hindi marks.
7. Use appropriate data visualization to plot the above Table (a) using the following charts
- Bar chart plot & pie plot
 - Histogram, Boxplot & QQPLOT
 - Dot plot, line chart, Scatterplot

8. Solve the following set of equations using Gaussian elimination method.

$$2x_1 + 5x_2 = 7$$

$$6x_1 + 12x_2 = 18.$$

9. Solve the following set of eqns using ~~Gaussian~~ ~~elimination~~ LU decomposition method $2x_1 + 5x_2 = 7$

$$6x_1 + 12x_2 = 18$$

10. Apply PCA for the following matrix & prove that it works

$$\begin{pmatrix} 4 & 3 \\ 1 & 2 \end{pmatrix}$$

11. Apply SVD for the following matrix $\begin{pmatrix} 4 & 3 \\ 1 & 2 \end{pmatrix}$

perform matrix decompositions & prove that SVD works.

12. Find covariance & correlation coefficients for the following 2 sets of data

$$X: 1 \quad 2 \quad 6 \quad 12$$

$$Y: 8 \quad 12 \quad 18 \quad 22$$

2.5 Basics of learning theory!

Learning is a process by which one can acquire knowledge & construct new ideas / concepts based on the experiences.

ML is an intelligent way of learning general concept from training examples without writing a program.

There are many ML algorithms through which computers can intelligently learn from past data / experiences, identify patterns & make predictions when new data is fed.

Computers can learn the tasks depending on the nature of the problems

There are 2 kinds of problems.

* Well posed

* Ill posed.

Computers can solve only well posed problems, as these have well defined specifications & have the following components inherent to it.

(1) class of learning Tasks (T)

(2) A measure of performance (P).

(3) A source of experience (E).

According to the Tom Mitchell, a program can learn from E for the Task T & P improves with experience E

Let x be the i/p & \mathcal{X} be the i/p space, which is the set of all inputs, & Y is the o/p space, which is the set of all possible o/p's i.e. Yes/No.

Let the unknown target function be $f: \mathcal{X} \rightarrow Y$, that maps the i/p space to o/p space

The objective of the learning program is to pick a function $g: \mathcal{X} \rightarrow Y$ to approximate hypothesis f .

All the possible formulae form a hypothesis space.

Let H be the set of all formulae from which the learning algorithm chooses.

The choice is good when the hypothesis g replicates f for all samples.

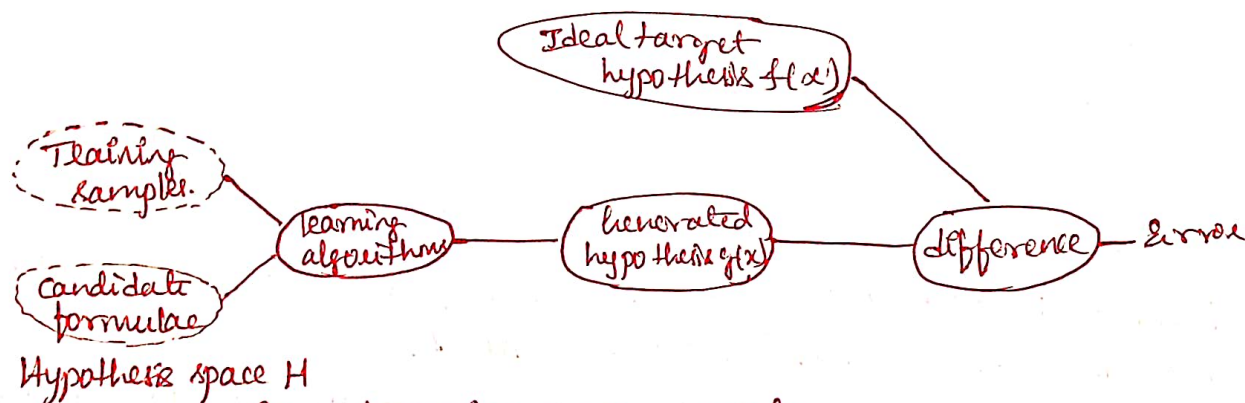


Fig: Learning environment.

Training samples & target function are dependent on the given problem.

The learning algorithm & hypothesis set are independent of the given problem.

Thus, the learning model is informally the hypothesis set & learning algorithm.

Thus, learning model can be stated as follows

Learning model = Hypothesis set + Learning algorithm

Let us assume a problem of predicting a label for a given i/p data.

Let ' D ' be the i/p dataset with both (+)ve & (-)ve example.

Let y be the o/p with class 0 (or) 1.

The simple learning model can be given as

$\sum_{i=1}^D x_i w_i > \text{Threshold}$, belongs to class 1 &

$\sum_{i=1}^D x_i w_i < \text{Threshold}$, belongs to another class.

This can be put into a single equation as follows

$$h(x) = \text{sign} \left(\left(\sum_{i=1}^D x_i w_i \right) + b \right)$$

Where x_1, x_2, \dots, x_D are the components of the i/p vector. w_1, w_2, \dots, w_D are the weights & $+1$ & -1 represent the class.

This simple model is called perception model.

$w_0 = b$ & fixing it as 1, then the model can further be simplified as

$$h(x) = \text{sign}(w^T x)$$

This is called perception learning algorithm.

Classical & Adaptive Machine Learning systems:-

A classical ML system has components such as i/p, process & o/p.

The i/p values are taken from the environment directly.

These values are processed & a hypothesis is generated as o/p model.

This model is then used for making predictions.

The predicted values are consumed by the environment.

Adaptive systems interact with the i/p for getting labelled data as direct i/p's are not available. This process is called reinforcement learning.

In reinforcement learning, a learning agent interacts with the environment & in return gets feedback.

Based on the feedback, the learning agent generates i/p samples for learning, which are used for generating the learning model. Such learning agents are not static & change their behaviour according to the external i/l received from the environment. This feedback is known as reward & learning.

Learning types:-

There are different types of learning. Some of the different learning methods are as follows.

Learn by memorization:-

Learn by repetition called rote learning is done by memorizing without understanding the logic/concept.

- * Learn by examples: / learn by experience / previous knowledge acquired at some time (Analogy).
- * Learn by being taught by an expert / a teacher, generally called a passive learning where the learner can interactively query a teacher / expert to label unlabelled data instances with the desired o/p.
- * Learn by critical thinking - Deductive learning, deduces new facts / conclusion from related ~~and~~ known facts & information
- * Self learning: Reinforcement learning - Normally learn from mistakes, punishments & rewards.
- * Learning to solve problems - Cognitive learning, where learning happens in the mind. & is possible by devising a methodology to achieve a goal
- * Learning by ~~solving problems~~ generalizing explanations, based learning (EBL) which exploits domain knowledge from experts to improve the accuracy of learned concepts by supervised learning.

Introduction to computational learning theory.

Computational learning theory (COLT) deals with formal methods used for learning systems.

It deals with frameworks for quantifying learning tasks & learning algorithms

It provides a fundamental basis for study of ML.

It deals with probability Approximate learning (PAC) & Vapnik-Chervonenkis (VC) dimensions

2.5. Design of a Learning system

A system that is built around a learning algorithm is called a learning system

The design of this focuses on these steps

1. Choosing a training experience
2. Choosing a target function
3. Representation of a target function
4. Function approximation

Training experience:-

Let us consider designing of a chess game.

In direct experience, individual board states & correct moves of the chess are given directly

In indirect system, the move sequences & results are only given

The training experience depends on the presence of a supervisor who can label all valid moves for a board state.

In the absence of a supervisor, the game agent plays against itself & learns the good moves, if the training samples cover all scenarios.

If the training samples & testing samples have the same distribution, the results would be good.

Determine the target function:

The next step is the determination of a target function.

In this step, the type of knowledge that needs to be learnt is determined

In direct experience, a board move is selected & it is determined whether it is a good move/not against all other moves.

If it is the best move, then it is chosen as $B \rightarrow M$, where B & M are legal moves

In indirect experience, all legal moves are accepted. & a score is generated for each.

The move with largest score is then chosen & executed

Determine the target function representation

The representation of knowledge may be a table, collection of rules, a neural n/w

The linear combination of these factors can be coined as

$$V = w_0 + w_1 x_1 + w_2 x_2 + w_3 x_3$$

where x_1, x_2, x_3 represent different board features & w_0, w_1, w_2, w_3 represent weights

Choosing an approximation algorithm for the target function.
The focus is to choose weights & fit the given training samples effectively.

The aim is to reduce the error given as

$$E = \sum_{\substack{\text{Training} \\ \text{Samples}}} [V_{\text{train}}(b) - \hat{V}(b)]^2$$

Here b is the sample

$\hat{V}(b)$ is the predicted hypothesis.

The approximation is carried out as

* Computing the error as the difference b/n trained & expected hypothesis
let error be $\text{error}(b)$.

* Then, for every board feature x_i , the weights are updated as
 $w_i = w_i + \mu \times \text{error}(b) \times x_i$

Here μ is the constant that moderates the size of the weight update

Thus, the learning system has the following components

- * A performance system to allow the game to play against itself
- * A critic system to generate the samples
- * A generalized. system to generate a hypothesis based on samples
- * An experimenter system to generate a new system based on the currently learnt function. This is sent as i/p to the performance sys.

Introduction to concept learning

It is a learning strategy of acquiring abstract knowledge | inferring a general concept | deriving a category from the given training samples
It is a process of abstraction & generalization from the data.

Concept learning requires 3 things.

- (1). Input: Training dataset which is a set of training instances, each labeled with the name of a concept / category to which it belongs. Use this past experience to train & build the model.
- (2). Output: Target concept / Target function f .
It is a mapping function $f(x)$ from \mathcal{X} to \mathcal{Y} .
- (3) Test: New instances to test the learned model.

Formally concept learning is defined as "Given a set of hypotheses, the learner searches through the Hypothesis space to identify the best hypothesis that matches the target concept."

Consider the following set of Training instances shown in Table 1.

Sl No	Horns	Tail	Tufts	Paws	Fur	Color	Hooves	Size	Elephant
1	No	Short	Yes	No	No	Black	No	Big	Yes
2	Yes	Short	No	No	No	Brown	Yes	Medium	No
3	No	Short	Yes	No	No	Black	No	Medium	Yes
4	No	Long	No	Yes	Yes	White	No	Medium	No
5	No	Short	Yes	Yes	Yes	Black	No	Big	Yes

Here, in this set of training instances, the independent attributes considered are "Horns", Tail, "Tufts", "Paws", "Fur", color, Hooves & 'Size'.

The dependent attribute is elephant.

The target concept is to identify the animal to be an elephant.

Representation of a Hypothesis:-

A hypothesis 'h' approximates a target function 'f' to represent the relationship b/w the independent attributes & the dependent attribute of the training instances.

The hypothesis is the predicted approximate model that best maps the \mathcal{X} to \mathcal{Y} .

Each hypothesis is represented as a conjunction of attribute conditions in the antecedent part

e.g.: $(\text{Tail} = \text{Short}) \wedge (\text{color} = \text{Black})$.

The set of hypotheses in the search space is called as hypothesis [singular form]

$H \rightarrow$ Hypotheses [plural form]

$h \rightarrow$ hypothesis [Candidate hypothesis]

Each attribute condition is the constraint on the attribute which is represented as attribute value pair.

In the antecedent of the attribute condition of a hypothesis, each attribute can take value as either ? or ψ or can hold a single value

* "?" denotes that the attribute can take any value [e.g. $\text{color} = ?$].

* " ψ " denotes that the attribute can't take any value i.e. it represents a null value [e.g. $\text{Horn} = \psi$]

* Single value denotes a specific single value from acceptable value of the attribute. i.e. the attribute "Tail" can take a value as "short"

e.g.: A hypothesis "h" will look like

	Horn	Tail	Tusks	Paws	Fur	color	Hooves	Size
$h =$	<No	?	Yes	?	?	Black	No	Medium>

Given a test instance x , we say $h(x) = 1$, if the test instance 'x' satisfies this hypothesis h .

Hypothesis space:

It is the set of all possible hypotheses that approximate the target function f .

The set of hypotheses that can be generated by a learning algorithm can be further reduced by specifying a language bias.

The subset of hypothesis space that is consistent with all observed training instances is called as version space.

Version space represents the only hypotheses that are used for the classification.

e.g., each of the attribute given in the table has the following possible set of values

Horns - Yes, No.

Tail - Long, short.

Tusk - Yes, No

Pauses - Yes, No

Fur - Yes, No

color - Brown, Black, White

Hooves - Yes, No

Size - Medium, Big

Considering these values for each of the attribute, there are $(2 \times 2 \times 2 \times 2 \times 2 \times 2 \times 2 \times 2) = 384$ distinct instances covering all the 5 instances in the training data set.

When 2 more values $[?, \psi]$ are added to each of the attribute, we can generate $(4 \times 4 \times 4 \times 4 \times 4 \times 5 \times 4 \times 4) = 81920$ distinct hypotheses.

Heuristic Space Search:-

This is a search strategy that finds an optimized hypothesis/solution to a problem by iteratively improving the hypothesis/solⁿ based on a given heuristic function or a cost measure.

Heuristic search methods will generate a possible hypothesis that can be a

path in the hypothesis space / a path from the initial state.

The hypothesis will be tested with the target function / the goal state to see if it is a real solⁿ.

If the tested hypothesis is a real solution, then it will be selected. This method generally increases the efficiency because it is guaranteed to find a better hypothesis, but may not be the best hypothesis.

Commonly used heuristic ^{Search} methods are

- * climbing methods.
- * Constraint satisfaction problems
- * Best-first search
- * Simulated-annealing.

Generalization & specialization

By generalization of the most specific hypothesis & by specialization of the most general hypothesis, the hypothesis space can be searched for an approximate hypothesis that matches all positive instances but does not match any negative instance.

Searching the hypothesis space:-

There are 2 ways of learning the hypothesis, consistent with all training instances from the large hypothesis space

1. Specialization - General to specific learning
2. Generalization - Specific to General learning.

Generalization - Specific to General learning:-

The learning methodology will search through the hypothesis space for an approximate hypothesis by generalizing the most-specific hypothesis.

Consider the training instances shown in table 1. & illustrate specific to General learning.

Set¹: we will start from all false / the most specific hypothesis to determine the most restrictive specialization
Consider only the positive instances & generalize the most hypothesis.

Ignore the negative instances.

(10)

This learning illustrated as follows

The most specific hypothesis is taken now, which will not classify any instance to true.

$h = \langle \psi \ \psi \ \psi \ \psi \ \psi \ \psi \ \psi \rangle$

Read the first instance I_1 , to generalize the hypothesis h so that this positive instance can be classified by the hypothesis h_1

I_1 : No Short Yes No No Black No Big Yes (Positive instance)

$h_1 = \langle \text{No Short Yes No No Black No Big} \rangle$

When reading the 2nd instance I_2 , it is a negative instance, so ignore it.

I_2 : Yes Short No No No Brown Yes Medium No (Negative instance)

$h_2 = \langle \text{No Short Yes No No Black No Big} \rangle$

Similarly, when reading the 3rd instance I_3 , it is a positive instance so generalize h_2 to h_3 to accommodate it. The resulting h_3 is generalized

I_3 : No Short Yes No No Black No Medium Yes (Positive instance)

$h_3 = \langle \text{No Short Yes No No Black No ?} \rangle$

Ignore I_4 since it is a Negative instance

I_4 : No Long No Yes Yes white No Medium No (Negative instance)

$h_4 = \langle \text{No Short Yes No No Black No ?} \rangle$

When reading the 5th instance I_5 , h_4 is further generalized to h_5 .

I_5 : No Short Yes Yes Yes Black No Big Yes (Positive instance)

$h_5 = \langle \text{No Short Yes ? ? Black No ?} \rangle$

after observing all the positive instances, an approximate hypothesis generated which can now classify any subsequent +ve instance

Specialization - General to specific learning!

This learning methodology will search through the hypothesis space for an approximate hypothesis, by specializing the most general hypothesis.

Illustrate learning by specialization - General to specific learning for the data instances shown in table 1.

Solⁿ:

Solⁿ: Start from the most general hypothesis which will make true all positive & negative instance.

Initially

$$h = \langle ? \quad ? \quad ? \quad ? \quad ? \quad ? \quad ? \quad ? \rangle$$

h is the more general to classify all instance to true

I_1 : No Short Yes No No Black No Big Yes (Give instance)

$$h_1 = \langle ? \quad ? \quad ? \quad ? \quad ? \quad ? \quad ? \quad ? \rangle$$

I_2 : Yes Short No No No Brown Yes Medium No (Give instance)

$$h_2 = \langle \text{No} \quad ? \quad ? \quad ? \quad ? \quad ? \quad ? \quad ? \rangle$$

$$\langle ? \quad ? \quad \text{Yes} \quad ? \quad ? \quad ? \quad ? \quad ? \rangle$$

$$\langle ? \quad ? \quad ? \quad ? \quad ? \quad \text{Black} \quad ? \quad ? \rangle$$

$$\langle ? \quad ? \quad ? \quad ? \quad ? \quad ? \quad \text{No} \quad ? \rangle$$

$$\langle ? \quad ? \quad ? \quad ? \quad ? \quad ? \quad ? \quad \text{Big} \rangle$$

h_2 imposes constraints so that it will not classify a Give

I_3 : No Short Yes No No Black No Medium Yes (true instance)

h_3 : \langle No ? ? ? ? ? ? \rangle

\langle ? ? Yes ? ? ? ? \rangle

\langle ? ? ? ? ? Black ? ? \rangle

\langle ? ? ? ? ? ? No ? \rangle

\langle ? ? ? ? ? ? ? Big \rangle

I_4 : No Long No Yes Yes white No Medium No (true instance)

h_4 : \langle ? ? Yes ? ? ? ? \rangle

\langle ? ? ? ? ? Black ? ? \rangle

\langle ? ? ? ? ? ? ? Big \rangle

Remove: any hypothesis inconsistent with this negative instance.

I_5 : No Short Yes Yes Yes Black No Big Yes (true instance)

h_5 : \langle ? ? Yes ? ? ? ? \rangle

\langle ? ? ? ? ? Black ? ? \rangle

\langle ? ? ? ? ? ? ? Big \rangle

Thus, h_5 is the hypothesis space generated which will classify the (true instances to true & negative instances to false

Hypothesis space search by Find S-algorithm

Find S-algorithm is guaranteed to converge to the most specific hypothesis in H that is consistent with the positive instances in the training dataset.

It considers only the +ve instances & eliminates negative instances while generating the hypothesis.

It initially starts with the most specific hypothesis.

Algorithm - To find 'S'

Input: positive instances in the training dataset.

Output: Hypothesis h .

(1). Initialize h to the most specific hypothesis.

$h = \langle \psi \ \psi \ \psi \ \psi \ \psi \ \dots \rangle$

(2) Generalize the initial hypothesis for the first positive instance

(3) For each subsequent instance,

If it is a positive instance,

check for each attribute value in the instance with the hypothesis h ,

If the attribute value is the same as the hypothesis value, then do nothing,

else if the attribute value is different than the hypothesis value, change it to '?' in h .

Else if it is a -ve instance,

ignore it.

Limitations of Find S-algorithm:-

1. Find S-algorithm tries to find a hypothesis that is consistent with positive instances, ignoring all -ve instances
2. The algorithm finds only one unique hypothesis, whereas there are many other hypotheses that are consistent with the training data.
3. Many times, the training dataset may contain some erroneous inconsistent data instances can mislead this algorithm in finding a consistent hypothesis since it ignores negative instances.

(19)

Apply the Find-S algorithm for the training dataset of 4 instances shown in table below

CRPA	Interactive needs	Practical knowledge	Communication Skills	Logical Thinking	Interest	Job Offer
>9	Yes	Excellent	Good	Fast	Yes	Yes
>9	Yes	Good	Good	Fast	Yes	Yes
>8	No	Good	Good	Fast	No	No
>9	Yes	Good	Good	Slow	No	Yes

Soln:

Step 1: Initialize 'h' is the most specific hypothesis. There are 6 attributes, so, for each attribute, we initially fill 'ψ' in the initial hypothesis 'h'.

$$h = \langle \psi \ \psi \ \psi \ \psi \ \psi \ \psi \rangle$$

Step 2: Generalize the initial hypothesis for the first positive instance.

I_1 is a positive instance, so generalize the most specific hypothesis 'h' to indicate this positive instance. Hence

I_1 : >9 Yes Excellent Good fast Yes +ve instance

$$h = \langle >9 \text{ Yes Excellent Good fast Yes} \rangle$$

Step 3: Scan the next instance I_2 , since I_2 is a +ve instance.

Generalize 'h' to include +ve instance I_2 .

For each of the non-matching attribute value in 'h' put a '?' to include this +ve instance.

The 3rd attribute value is mismatching in 'h' with I_2 , so put a '?'

I_2 : >9 Yes Good Good fast Yes +ve instance

$$h = \langle >9 \text{ Yes ? Good fast Yes} \rangle$$

Now Scan I_3 , since it is a +ve instance, ignore it. Hence, the hypothesis remain the same without any change after scanning I_3 .

I_3 : >8 No Good Good fast No Negative instance

$$h = \langle >9 \text{ Yes ? Good fast Yes} \rangle$$

Scan I_4 , Since it is a +ve instance, check for mismatch in the hypothesis

I_4 :

4th attribute value are mismatching, so add ? to those attributes in h

Yes Good Good Slow No Positive instance

$$h = \langle >9 \text{ ? Good ? ?} \rangle$$

hypothesis generalized with Find S-algorithm is

$$h = \langle >9 \text{ ? Good ? ?} \rangle$$

It positive instances & ignores any -ve instances