Unit III

Rule Induction - Neural Networks: Learning and Generalization - Competitive Learning - Principal Component Analysis and Neural Networks - Fuzzy Logic: Extracting Fuzzy Models from Data - Fuzzy Decision Trees - Stochastic Search Methods- Neuro-Fuzzy Modelling – Association rule mining – Clustering – Outlier Analysis – Sequential Pattern Mining – Temporal mining – Spatial mining – Web mining

NEURAL NETWORK

A **neural network** is used to refer to a network of biological neurons. A neural network consists of a set of highly interconnected entities called nodes or units. Each unit accepts a weighted set of inputs and responds with an output.

Neural networks are a class of learning algorithms that employ a "network" of interconnected nodes in various layers. It's based on the animal nervous system, with nodes resembling neurons and edges resembling synapses. The network establishes computational rules for passing input data from the network's input layer to the network's output layer, and each edge has an associated weight.

The relationship between the input and output layers, which is parameterized by the weights, is described by a network function associated with a neural network.

Mathematically let $I = (I_1, I_2, \dots \dots I_n)$ represent the set of inputs presented to the unit U. Each input has an associated weight that represents the strength of that particular connection. Let $W = (W_1, W_2, \dots \dots W_n)$ represent the weight vector corresponding to the input vector X. By applying to
V, these weighted inputs produce a net sum at U given by

$$S = \text{SUM}(W_i \times I_i)$$

Neural network structures

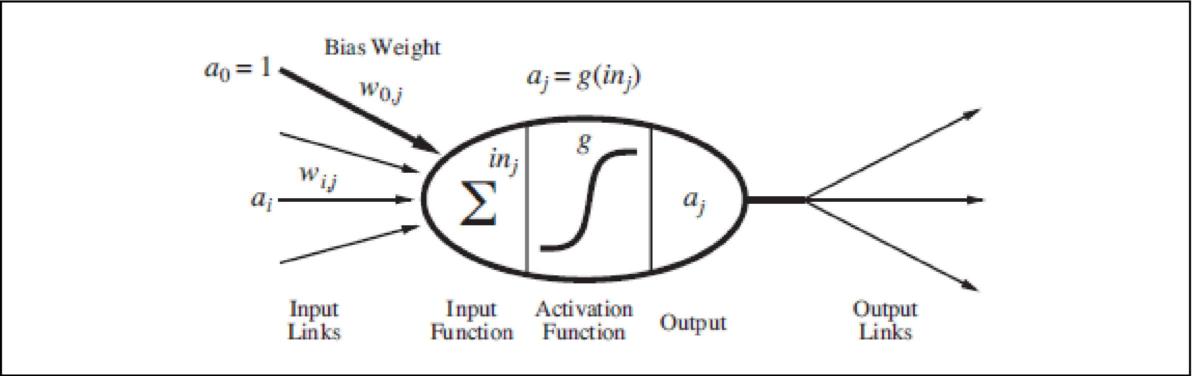


Figure 18.19 A simple mathematical model for a neuron. The unit's output activation is $a_j = g(\sum_{i=0}^{n} w_{i,j} a_i)$, where a_i is the output activation of unit i and $w_{i,j}$ is the weight on the link from unit i to this unit.

Neural networks are composed of nodes or **units** (see Figure 18.19) connected by directed **links**. A link from unit i to unit j serves to propagate the **activation** a_i from i to j.[8] Each link also has a numeric **weight** $w_{i,j}$ associated with it, which determines the strength and sign of the connection. Just as in linear regression models, each unit has a dummy input $a_0 = 1$ with an associated weight $w_{0,j}$. Each unit j first computes a weighted sum of its inputs:

$$in_j = \sum_{i=0}^{n} w_{i,j} a_i \ .$$

Then it applies an **activation function** g to this sum to derive the output:

$$a_j = g(in_j) = g\left(\sum_{i=0}^{n} w_{i,j} a_i\right) \ . \tag{18.9}$$

The activation function g is typically either a hard threshold (Figure 18.17(a)), in which case the unit is called a **perceptron**, or a logistic function (Figure 18.17(b)), in which case the term **sigmoid perceptron** is sometimes used. Both of these nonlinear activation function ensure the important property that the entire network of units can represent a nonlinear function

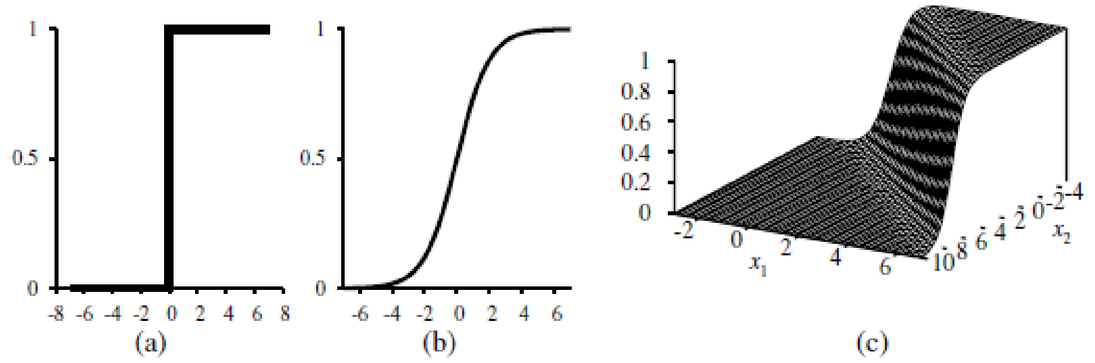
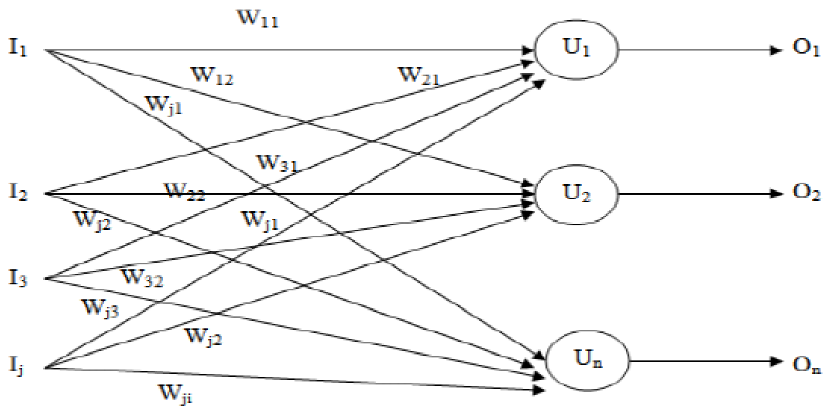


Figure 18.17 (a) The hard threshold function $Threshold(z)$ with 0/1 output. Note that the function is nondifferentiable at $z=0$. (b) The logistic function, $Logistic(z) = \frac{1}{1+e^{-z}}$, also known as the sigmoid function. (c) Plot of a logistic regression hypothesis $h_{\mathbf{w}}(\mathbf{x}) = Logistic(\mathbf{w} \cdot \mathbf{x})$ for the data shown in Figure 18.15(b).

TYPES OF NEURAL NETWORKS

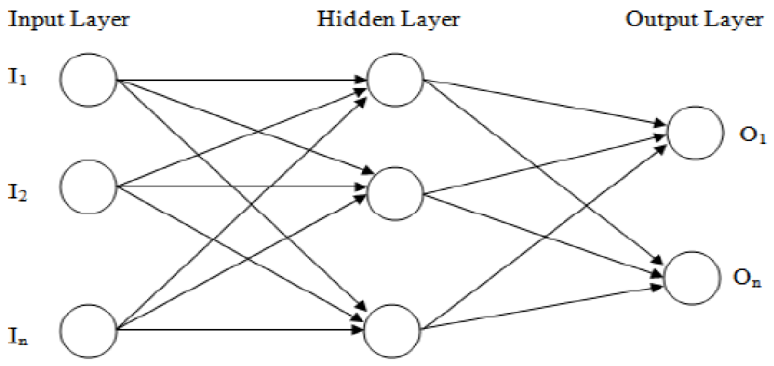
Single Layer Network: *A single layer neural network consists of a set of units organized in a layer. Each unit U_n receives a weighted input I_j with weight W_{jn}. Figure shows a single layer neural network with j inputs and outputs.*



Single Layer neural Network

Multilayer Network

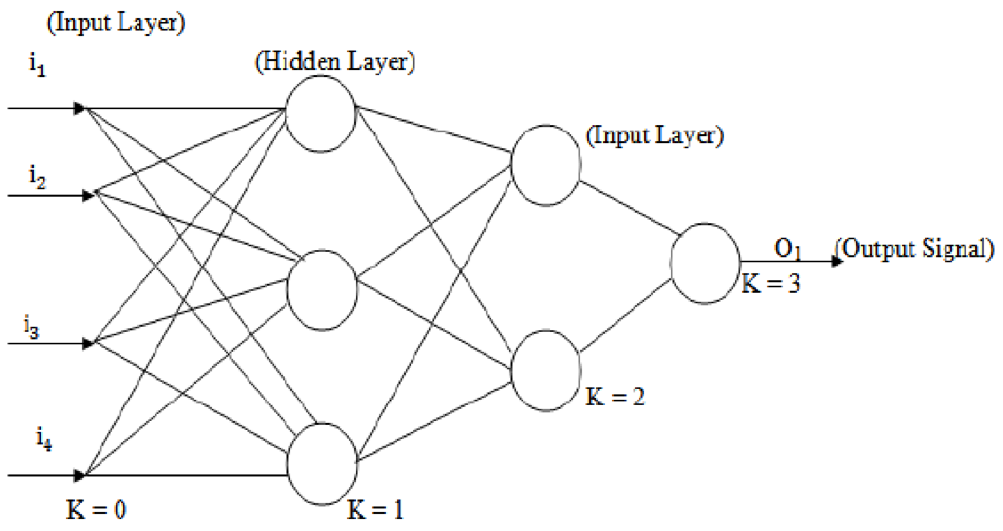
A multilayer network has two or more layers of units, with the output from one layer serving as input to the next. Generally in a multilayer network there are 3 layers present like, input layer, output layer and hidden layer. The layer with no external output connections are referred to as hidden layers. A multilayer neural network structure is given in figure.



A multilayer neural network

Feed Forward neural network

In this network, the information moves in only one direction, forward from the input nodes, through the hidden nodes and to the output nodes. There are no cycles or loops in the network. In other way we can say the feed forward neural network is one that does not have any connections from output to input. All inputs with variable weights are connected with every other node. A single layer feed forward network has one layer of nodes, whereas a multilayer feed forward network has multiple layers of nodes. The structure of a feed forward multilayer network is given in figure.



A **feed-forward network** has connections only in one direction—that is, it forms a directed acyclic graph. Every node receives input from "upstream" nodes and delivers output to "downstream" nodes; there are no loops. A feed-forward network represents a function of its current input; thus, it has no internal state other than the weights themselves.

Feed-forward networks are usually arranged in **layers**, such that each unit receives input only from units in the immediately preceding layer.

Types of feed forward Neural Network. They are

(i) Single Layer feed forward Network

- ✓ This is the simplest type of feed forward neural network because it only has one layer of neurons.
- ✓ It is often used for simple classification tasks.
- ✓ The input goes through the neurons in the output layer, where each neuron uses a linear or nonlinear activation function to make an output.
- ✓ The final output of the network is then made by adding up all the outputs of the neurons.
- ✓ One advantage is that they are computationally efficient and can be trained quickly.
- ✓ This network is simple and efficient.
- ✓ They have limitations regarding their ability to model complex data

Single-layer feed-forward neural networks (perceptrons)

A network with all the inputs connected directly to the outputs is called a **single-layer neural network**, or a **perceptron network**. Figure 18.20 shows a simple two-input, two-output perceptron network. With such a network, we might hope to learn the two-bit adder function, for example. Here are all the training data we will need:

x_1	x_2	y_3 (carry)	y_4 (sum)
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

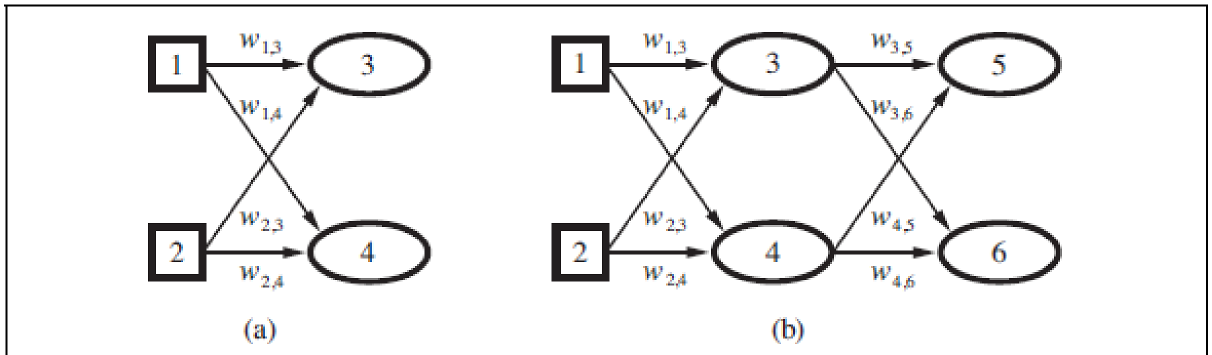


Figure 18.20 (a) A perceptron network with two inputs and two output units. (b) A neural network with two inputs, one hidden layer of two units, and one output unit. Not shown are the dummy inputs and their associated weights.

(ii) Multi Layer feed forward Network

- ✓ A multi-layer feed forward neural network (MFNN), is also called a multi-layer perceptron (MLP)
- ✓ This type of network has one or more hidden layers between the input and output layers.
- ✓ Data is sent to the input layer, which processes nonlinear activation functions through the hidden layers.
- ✓ The outputs of the hidden layers are then passed to the output layer, which produces the network's output.
- ✓ MFNNs are more potent than single-layer feed forward networks because they can learn complex nonlinear mappings between the input and output data.
- ✓ The hidden layers let the network learn more complicated ways to represent the data it gets, which makes it better at solving complex problems.

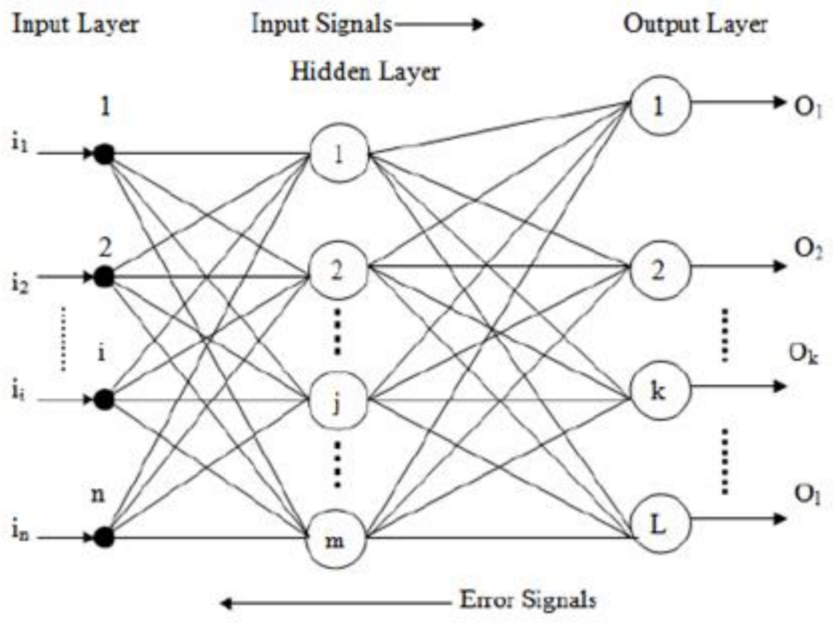
- ✓ This network needs lot of data and processing power.

(iii) Convolutional Neural Network (CNN)

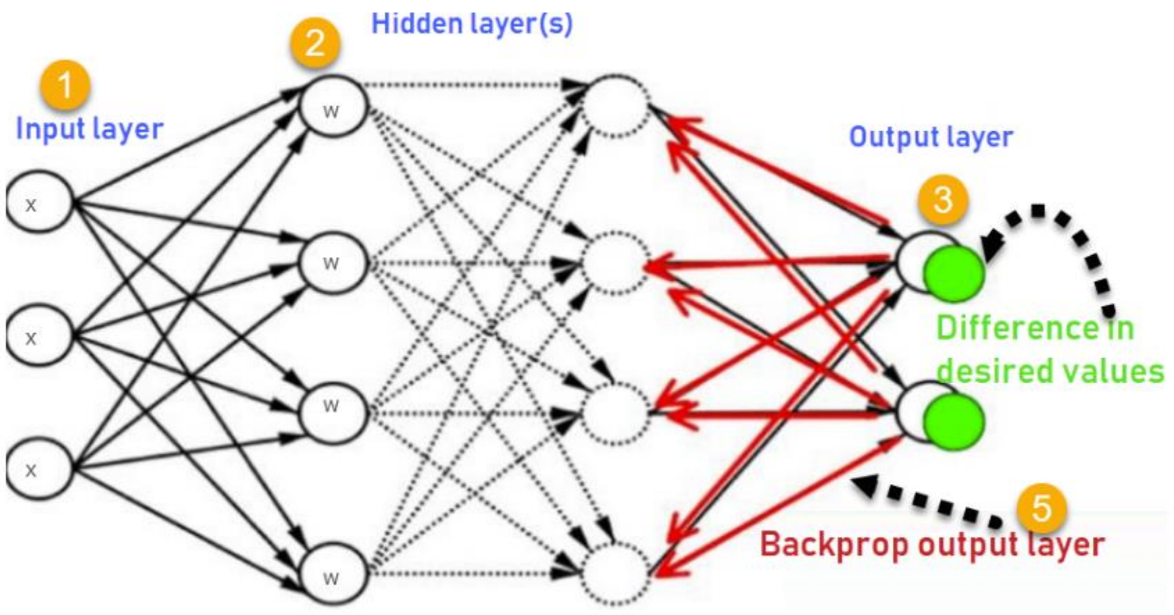
- ✓ This type of network is often used for image recognition and computer vision tasks.
- ✓ Used for object detection and image classification.
- ✓ It uses convolutional layers to find features in the image it is given and pool layers to reduce the number of dimensions in the image it gives back.

(iv) Back Propagation Neural Network

Multilayer neural networks use a most common technique from a variety of learning technique, called the back propagation algorithm. In back propagation neural network, the output values are compared with the correct answer to compute the value of some predefined error function. By various techniques the error is then fed back through the network. Using this information, the algorithms adjust the weights of each connection in order to reduce the value of the error function by some small amount. After repeating this process for a sufficiently large number of training cycles the network will usually converge to some state where the error of the calculation is small.



A recurrent network, on the other hand, feeds its outputs back into its own inputs. This means that the activation levels of the network form a dynamical system that may reach a stable state or exhibit oscillations or even chaotic behavior. Moreover, the response of the network to a given input depends on its initial state, which may depend on previous inputs. Hence, recurrent networks (unlike feed-forward networks) can support short-term memory.



The goal of back propagation, as with most training algorithms, is to iteratively adjust the weights in the network to produce the desired output by minimizing the output error. The algorithm's goal is to solve credit assignment problem. Back propagation is a gradient-descent approach in that it uses the minimization of first-order derivatives to find an optimal solution. The standard back propagation algorithm is given below.

Step1: Build a network with the choosen number of input, hidden and output u n i t s .
Step2: Initialize all the weights to low random values.

Step3: Randomly, choose a single training pair.

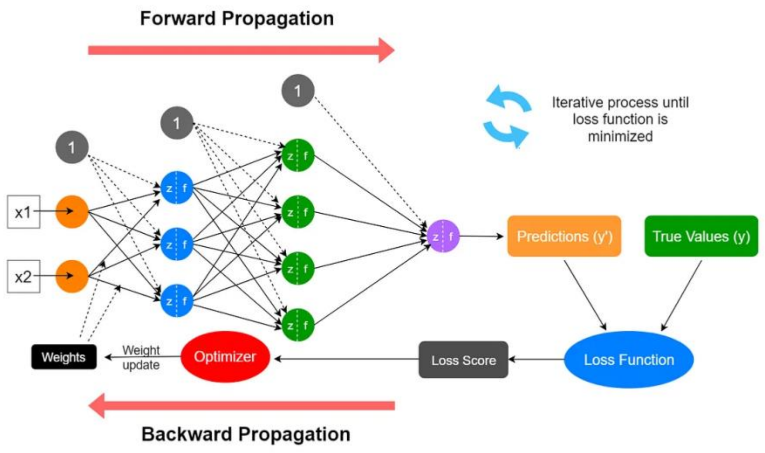
Step4: Copy the input pattern to the input layer.

Step5: Cycle the network so that the activation from the inputs generates the activations in the hidden and output layers.

Step6: Calculate the error derivative between the output activation and the final o u t p u t .

Step7: Apply the method of back propagation to the summed products of the weights and errors in the output layer in order to calculate the error in the hidden units.

Step8:Update the weights attached the each unit according to the error in that unit, the output from the unit below it and the learning parameters, until the error is sufficiently low.



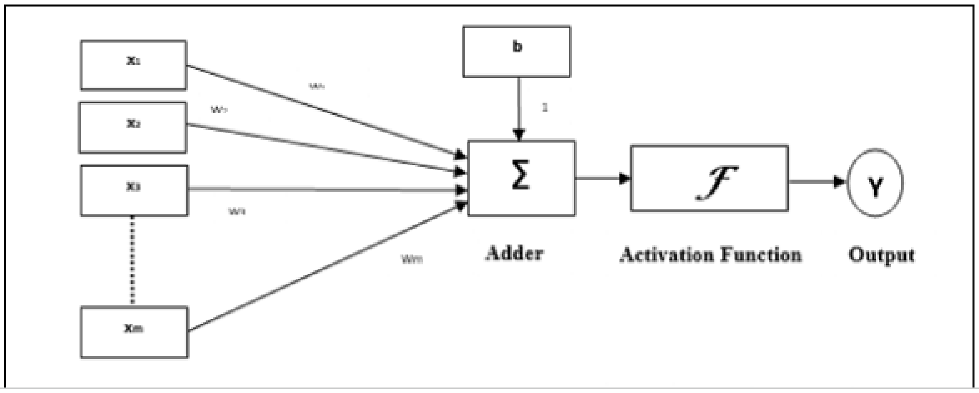
Different learning rules in the Neural network:

- Hebbian learning rule – It identifies, how to modify the weights of nodes of a network.
- Perceptron learning rule – Network starts its learning by assigning a random value to each weight.
- Delta learning rule – Modification in sympatric weight of a node is equal to the multiplication of error and the input.
- Correlation learning rule – The correlation rule is the supervised learning.
- Outstar learning rule – We can use it when it assumes that nodes or neurons in a network arranged in a layer.

Perceptron

Developed by Frank Rosenblatt by using McCulloch and Pitts model, perceptron is the basic operational unit of artificial neural networks. It employs supervised learning rule and is able to classify the data into two classes.

Operational characteristics of the perceptron: It consists of a single neuron with an arbitrary number of inputs along with adjustable weights, but the output of the neuron is 1 or 0 depending upon the threshold. It also consists of a bias whose weight is always 1. Following figure gives a schematic representation of the perceptron.



- **Links** − It would have a set of connection links, which carries a weight including a bias always having weight 1.
- **Adder** − It adds the input after they are multiplied with their respective weights.
- **Activation function** − It limits the output of neuron. The most basic activation function is a Heaviside step function that has two possible outputs. This function returns 1, if the input is positive, and 0 for any negative input.

Training Algorithm

Perceptron network can be trained for single output unit as well as multiple output units.

Training Algorithm for Single Output Unit

Step 1 − Initialize the following to start the training −

- Weights
- Bias
- Learning rate α

For easy calculation and simplicity, weights and bias must be set equal to 0 and the learning rate must be set equal to 1.

Step 2 − Continue step 3-8 when the stopping condition is not true.

Step 3 − Continue step 4-6 for every training vector **x**.

Step 4 − Activate each input unit as follows −

$$x_i \;=\; s_i \;(i \;=\; 1 \, to \, n)$$

Step 5 − Now obtain the net input with the following relation −

$$y_{in} \;=\; b \;+\; \sum_i^n x_i.\; w_i$$

Here **'b'** is bias and **'n'** is the total number of input neurons.

Step 6 − Apply the following activation function to obtain the final output.

$$f(y_{in}) = \begin{cases} 1 & if\, y_{in} > \theta \\ 0 & if\, -\theta \leqslant y_{in} \leqslant \theta \\ -1 & if\, y_{in} < -\theta \end{cases}$$

Step 7 − Adjust the weight and bias as follows −

Case 1 − if **y** ≠ **t** then,

$$w_i(new) = w_i(old) + \alpha\, t x_i$$

$$b(new) = b(old) + \alpha t$$

Case 2 − if **y** = **t** then,

$$w_i(new) = w_i(old)$$

$$b(new) = b(old)$$

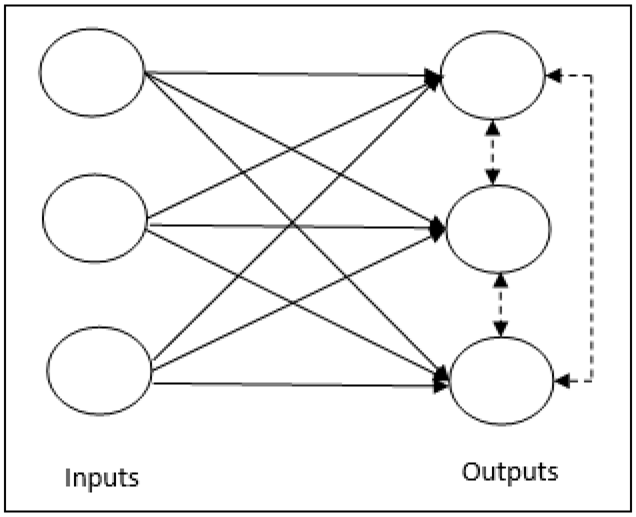
Here **'y'** is the actual output and **'t'** is the desired/target output.

Step 8 − Test for the stopping condition, which would happen when there is no change in weight.

Competitive Learning Rule (Winner-takes-all)

It is concerned with unsupervised training in which the output nodes try to compete with each other to represent the input pattern. To understand this learning rule, we must understand the competitive network which is given as follows −

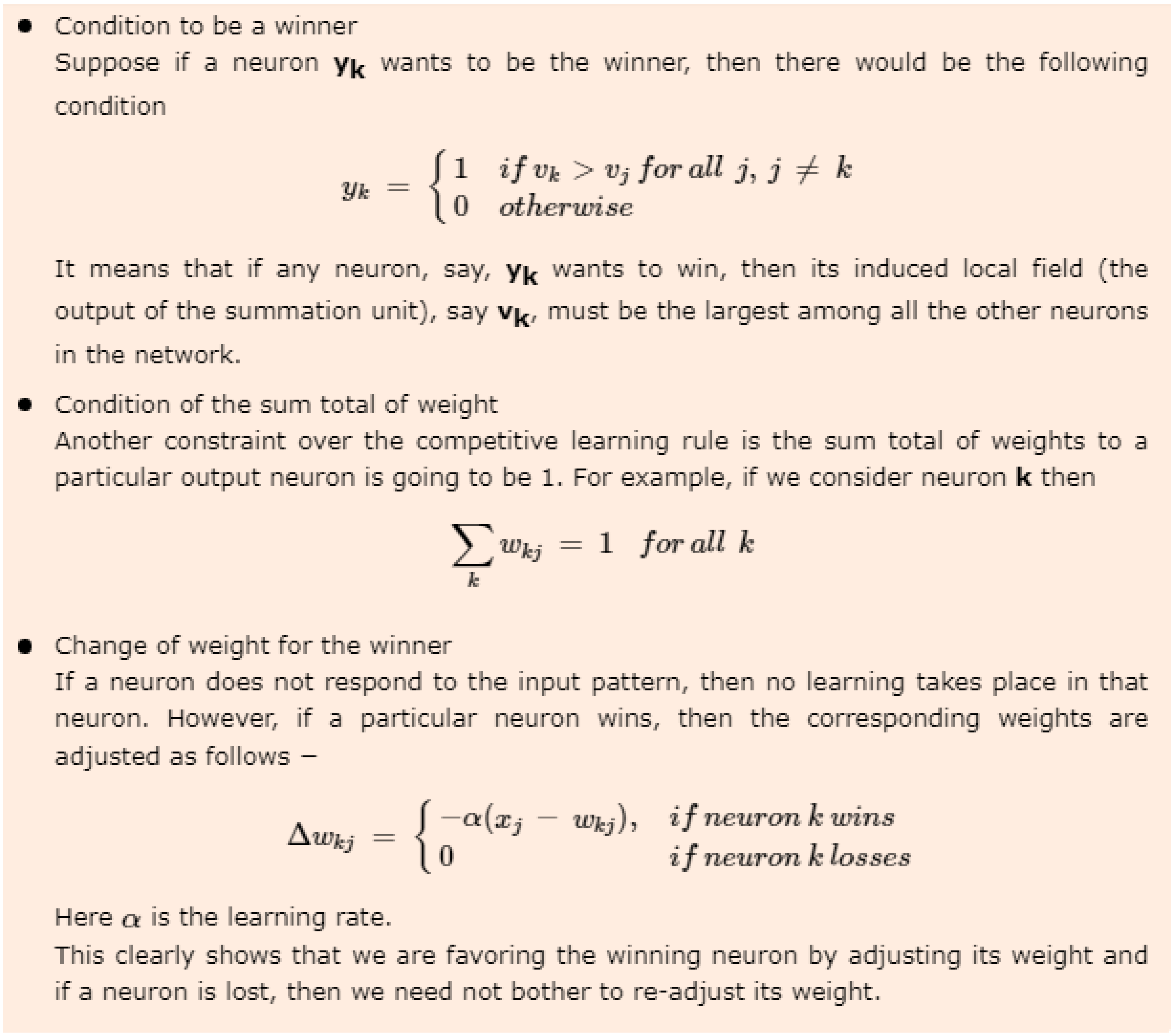
Basic Concept of Competitive Network − This network is just like a single layer feedforward network with feedback connection between outputs. The connections between outputs are inhibitory type, shown by dotted lines, which means the competitors never support themselves.



Competitive network

Basic Concept of Competitive Learning Rule − As said earlier, there will be a competition among the output nodes. Hence, the main concept is that during training, the output unit with the highest activation to a given input pattern, will be declared the winner. This rule is also called Winner-takes-all because only the winning neuron is updated and the rest of the neurons are left unchanged.

Mathematical formulation − Following are the three important factors for mathematical formulation of this learning rule −



- Condition to be a winner
  Suppose if a neuron $\mathbf{y_k}$ wants to be the winner, then there would be the following condition

$$y_k = \begin{cases} 1 & if\, v_k > v_j\, for\, all\ j,\, j \neq k \\ 0 & otherwise \end{cases}$$

It means that if any neuron, say, $\mathbf{y_k}$ wants to win, then its induced local field (the output of the summation unit), say $\mathbf{v_k}$, must be the largest among all the other neurons in the network.

- Condition of the sum total of weight
  Another constraint over the competitive learning rule is the sum total of weights to a particular output neuron is going to be 1. For example, if we consider neuron **k** then

$$\sum_k w_{kj} = 1 \quad for\, all\ k$$

- Change of weight for the winner
  If a neuron does not respond to the input pattern, then no learning takes place in that neuron. However, if a particular neuron wins, then the corresponding weights are adjusted as follows −

$$\Delta w_{kj} = \begin{cases} -\alpha(x_j - w_{kj}), & if\, neuron\, k\, wins \\ 0 & if\, neuron\, k\, losses \end{cases}$$

Here α is the learning rate.
This clearly shows that we are favoring the winning neuron by adjusting its weight and if a neuron is lost, then we need not bother to re-adjust its weight.

Generalization examines how well a model can digest new data and make correct predictions after getting trained on a training set.
If you train a model too well on training data, it will be incapable of generalizing. In such cases, it will end up making erroneous predictions when it's given new data. This would make the model ineffective even though it's capable of making correct predictions for the training data set. This is known as overfitting.

Fuzzy Logic

Fuzzy Logic Systems (FLS) produce acceptable but definite output in response to incomplete, ambiguous, distorted, or inaccurate (fuzzy) input.

What is Fuzzy Logic?

Fuzzy Logic (FL) is a method of reasoning that resembles human reasoning. The approach of FL imitates the way of decision making in humans that involves all intermediate possibilities between digital values YES and NO. In Fuzzy Logic, the degree of truth is between 0 and 1.

The conventional logic block that a computer can understand takes precise input and produces a definite output as TRUE or FALSE, which is equivalent to human's YES or NO.

The fuzzy logic works on the levels of possibilities of input to achieve the definite output.
Example: William is smart (0.8 truth)

Implementation

- It can be implemented in systems with various sizes and capabilities ranging from small micro-controllers to large, networked, workstation-based control systems.
- It can be implemented in hardware, software, or a combination of both.

Why Fuzzy Logic?

Fuzzy logic is useful for commercial and practical purposes.

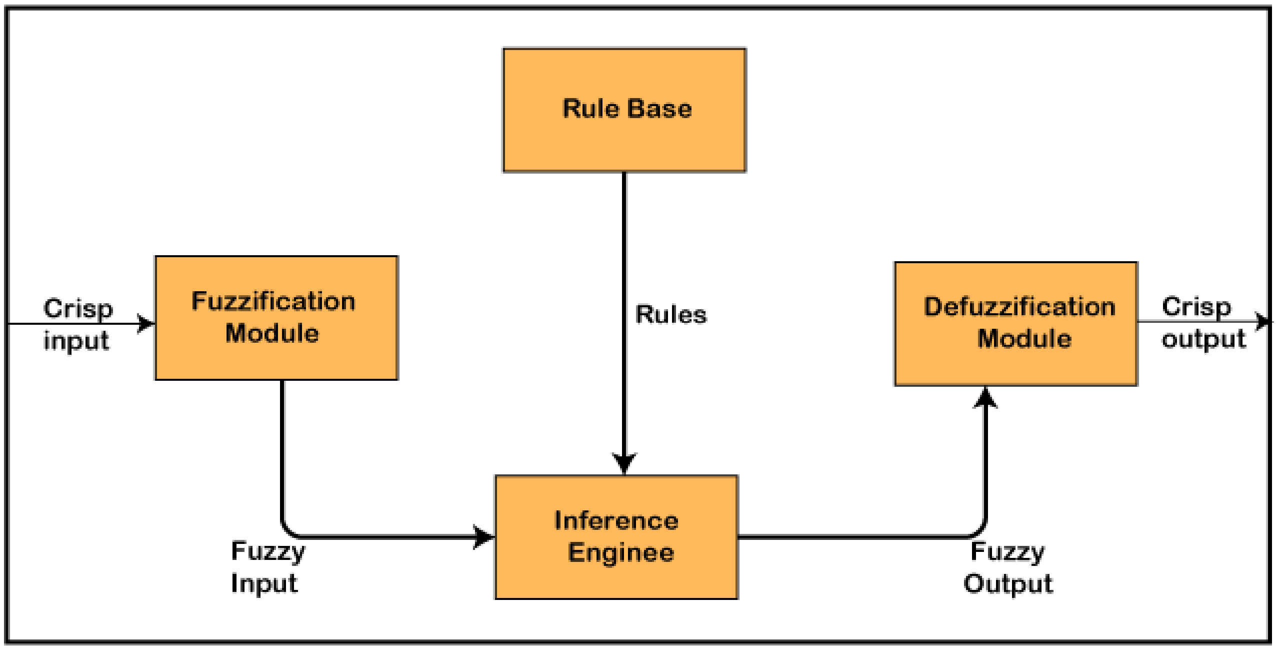
- It can control machines and consumer products.
- It may not give accurate reasoning, but acceptable reasoning.
- Fuzzy logic helps to deal with the uncertainty in engineering.

Architecture of a Fuzzy Logic System

In the architecture of the **Fuzzy Logic** system, each component plays an important role. The architecture consists of the different four components which are given below.

1. Rule Base
2. Fuzzification
3. Inference Engine
4. Defuzzification

Following diagram shows the architecture or process of a Fuzzy Logic system:

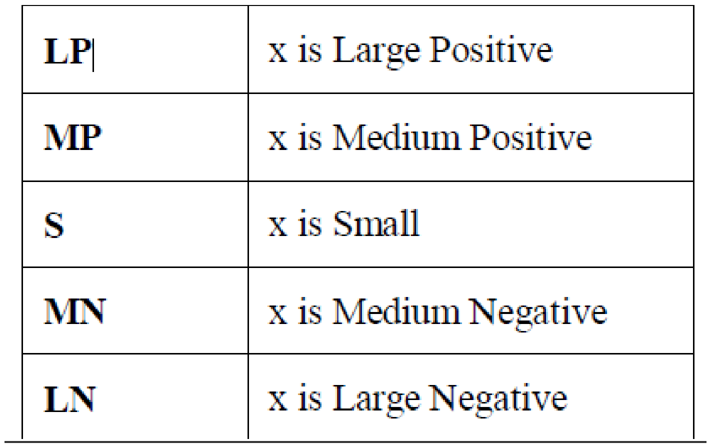


(i) Rule Base

Rule Base is a component used for storing the set of rules and the If-Then conditions given by the experts are used for controlling the decision-making systems. There are so many updates that come in the Fuzzy theory recently, which offers effective methods for designing and tuning of fuzzy controllers. These updates or developments decreases the number of fuzzy set of rules. This Knowledge Base stores IF_THEN rules provided by experts.

(ii) Fuzzification Module

Fuzzification is a module or component for transforming the system inputs, i.e., it converts the crisp number into fuzzy sets. The crisp numbers are those inputs which are measured by the sensors and then fuzzification passed them into the control systems for further processing. This component divides the input signals into following five states in any Fuzzy Logic system:



LP	x is Large Positive
MP	x is Medium Positive
S	x is Small
MN	x is Medium Negative
LN	x is Large Negative

(iii) Inference Engine

This component is a main component in any Fuzzy Logic system (FLS), because all the information is processed in the Inference Engine. It allows users to find the matching degree between the current fuzzy input and the rules. After the matching degree, this system determines which rule is to be added according to the given input field. When all rules are fired, then they are combined for developing the control actions. Inference Engine simulates the human reasoning process by making fuzzy inference on the inputs and IF_THEN rules.

(iv) Defuzzification Module

This is a module or component, which takes the fuzzy set inputs generated by the **Inference Engine**, and then transforms them into a crisp value. It is the last step in the process of a fuzzy logic system. The crisp value is a type of value which is acceptable by the user. Various techniques are present to do this, but the user has to select the best one for reducing the errors.

Membership Function

The membership function is a function which represents the graph of fuzzy sets, and allows users to quantify the linguistic term. It is a graph which is used for mapping each element of x to the value between 0 and 1.

This function is also known as indicator or characteristics function.

For the Fuzzy set B, the membership function for X is defined as: μB:X → [0,1]. In this function X, each element of set B is mapped to the value between 0 and 1. This is called a degree of membership or membership value

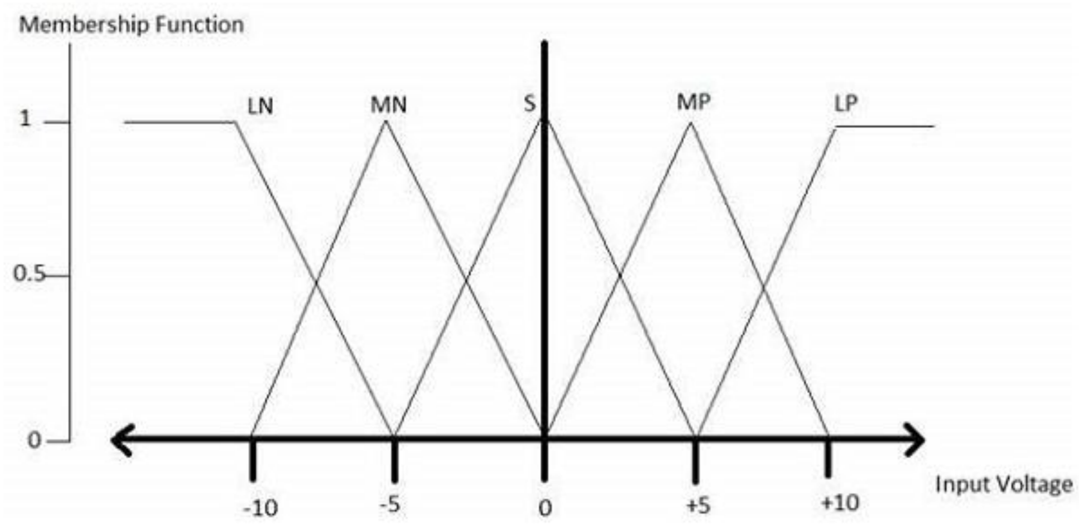
The **membership functions work on** fuzzy sets of variables. Membership functions allow you to quantify linguistic term and represent a fuzzy set graphically. A **membership function** for a fuzzy *set A* on the universe of discourse X is defined as μ_A: X → [0,1].

Here, each element of *X* is mapped to a value between 0 and 1. It is called **membership value** or **degree of membership**. It quantifies the degree of membership of the element in *X* to the fuzzy set *A*.

- x axis represents the universe of discourse.
- y axis represents the degrees of membership in the [0, 1] interval.

There can be multiple membership functions applicable to fuzzify a numerical value. Simple membership functions are used as use of complex functions does not add more precision in the output.

All membership functions for **LP, MP, S, MN,** and **LN** are shown as below –

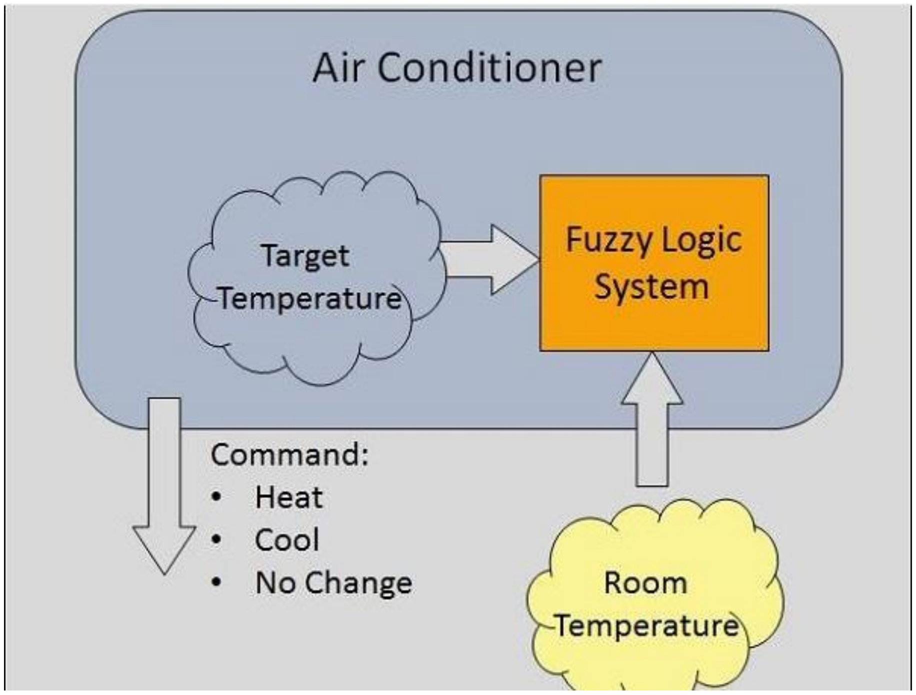


The triangular membership function shapes are most common among various other membership function shapes such as trapezoidal, singleton, and Gaussian.

Here, the input to 5-level fuzzifier varies from -10 volts to +10 volts. Hence the corresponding output also changes.

Example of a Fuzzy Logic System

Let us consider an air conditioning system with 5-level fuzzy logic system. This system adjusts the temperature of air conditioner by comparing the room temperature and the target temperature value.



Algorithm

- Define linguistic Variables and terms (start)
- Construct membership functions for them. (start)
- Construct knowledge base of rules (start)
- Convert crisp data into fuzzy data sets using membership functions. (fuzzification)
- Evaluate rules in the rule base. (Inference Engine)
- Combine results from each rule. (Inference Engine)
- Convert output data into non-fuzzy values. (defuzzification)

Development

Step 1 – Define linguistic variables and terms

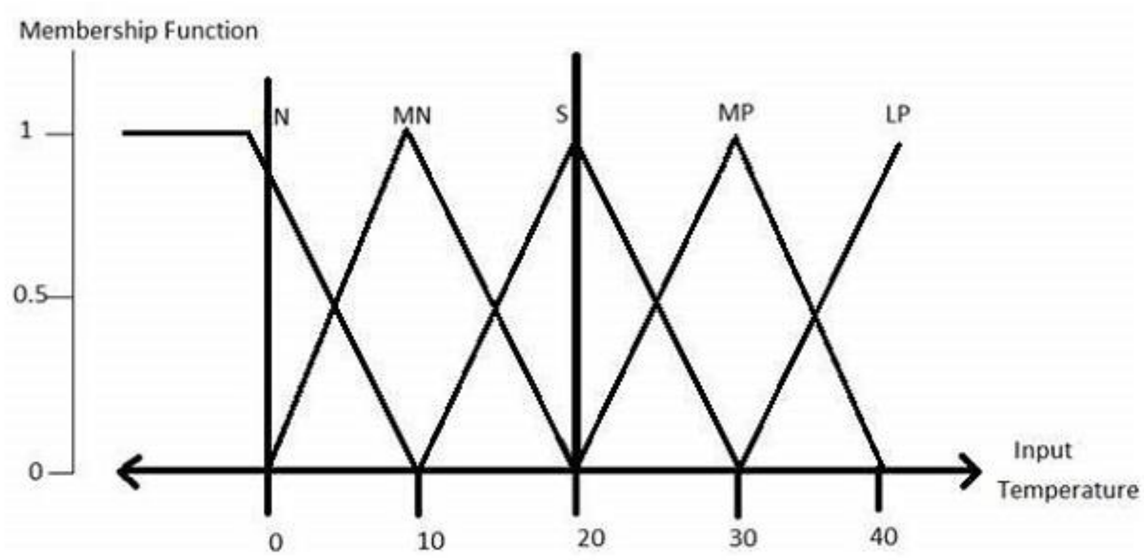
Linguistic variables are input and output variables in the form of simple words or sentences. For room temperature, cold, warm, hot, etc., are linguistic terms.

Temperature (t) = {very-cold, cold, warm, very-warm, hot}

Every member of this set is a linguistic term and it can cover some portion of overall temperature values.

Step 2 – Construct membership functions for them

The membership functions of temperature variable are as shown –



Step3 – Construct knowledge base rules

Create a matrix of room temperature values versus target temperature values that an air conditioning system is expected to provide.

RoomTemp. /Target	Very_Cold	Cold	Warm	Hot	Very_Hot
Very_Cold	No_Change	Heat	Heat	Heat	Heat
Cold	Cool	No_Change	Heat	Heat	Heat
Warm	Cool	Cool	No_Change	Heat	Heat
Hot	Cool	Cool	Cool	No_Change	Heat
Very_Hot	Cool	Cool	Cool	Cool	No_Change

Build a set of rules into the knowledge base in the form of IF-THEN-ELSE structures.

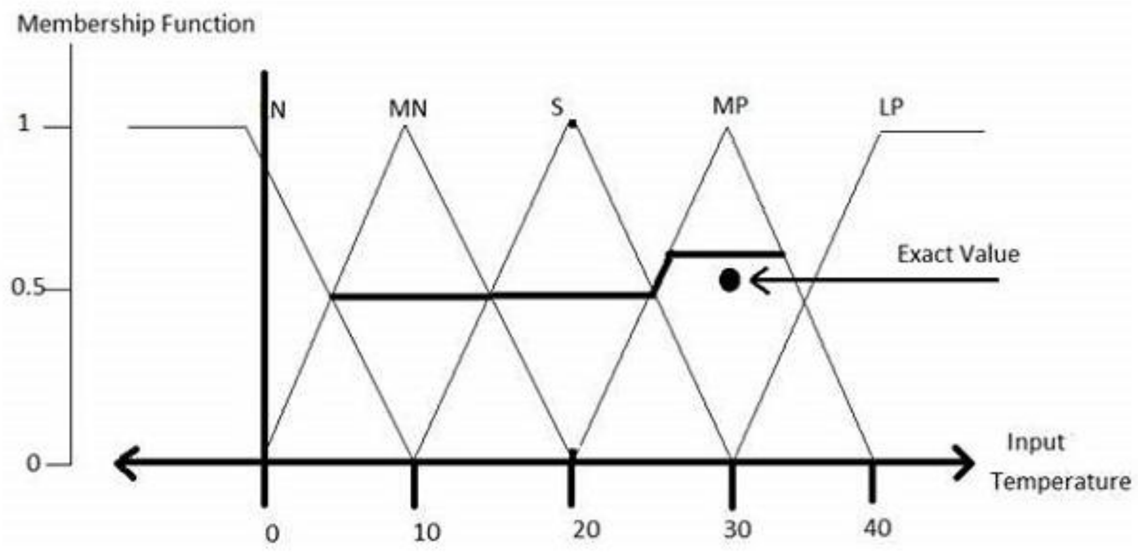
Sr. No.	Condition	Action
1	IF temperature=(Cold OR Very_Cold) AND target=Warm THEN	Heat
2	IF temperature=(Hot OR Very_Hot) AND target=Warm THEN	Cool
3	IF (temperature=Warm) AND (target=Warm) THEN	No_Change

Step 4 – Obtain fuzzy value

Fuzzy set operations perform evaluation of rules. The operations used for OR and AND are Max and Min respectively. Combine all results of evaluation to form a final result. This result is a fuzzy value.

Step 5 – Perform defuzzification

Defuzzification is then performed according to membership function for output variable.



Association Rule Mining

Association rule mining is a technique used to uncover hidden relationships between variables in large datasets. It is a popular method in data mining and machine learning and has a wide range of applications in various fields, such as **market basket analysis**, customer segmentation, and fraud detection.

What is Association Rule Mining?

Association rule mining is a technique used to identify patterns in large data sets. It involves finding relationships between variables in the data and using those relationships to make predictions or decisions. The goal of association rule mining is to uncover rules that describe the relationships between different items in the data set.

For example, consider a dataset of transactions at a grocery store. Association rule mining could be used to identify relationships between items that are frequently purchased together. For example, the rule "If a customer buys bread, they are also likely to buy milk" is an association rule that could be mined from this data set. We can use such rules to inform decisions about store layout, product placement, and marketing efforts.

Association rule mining typically involves using algorithms to analyze the data and identify the relationships. These algorithms can be based on statistical methods or machine learning techniques. The resulting rules are often expressed in the form of "if-then" statements, where the "if" part represents the antecedent (the condition being tested) and the "then" part represents the consequent (the outcome that occurs if the condition is met).

Association rule mining is an important technique in data analysis because it allows users to discover patterns or relationships within data that may not be immediately apparent. By identifying associations between variables, association rule mining can help users understand the relationships between different variables and how those variables may be related to one another.

This can be useful for various purposes, such as identifying market trends, detecting fraudulent activity, or understanding customer behavior. Association rule mining can also be used as a stepping stone for other types of data analysis, such as predicting outcomes or identifying key drivers of certain phenomena. Overall, association rule mining is a valuable tool for extracting insights and understanding the underlying structure of data.

Use Cases of Association Rule Mining:

Association rule mining is commonly used in a variety of applications, some common ones are:

Market Basket Analysis

One of the most well-known applications of association rule mining is in market basket analysis. This involves analyzing the items customers purchase together to understand their purchasing habits and preferences.

For example, a retailer might use association rule mining to discover that customers who purchase diapers are also likely to purchase baby formula. We can use this information to optimize product placements and promotions to increase sales.

Customer Segmentation

Association rule mining can also be used to segment customers based on their purchasing habits.

For example, a company might use association rule mining to discover that customers who purchase certain types of products are more likely to be younger. Similarly, they could learn that customers who purchase certain combinations of products are more likely to be located in specific geographic regions.

We can use this information to tailor marketing campaigns and personalized recommendations to specific customer segments.

Fraud Detection

You can also use association rule mining to detect fraudulent activity. For example, a credit card company might use association rule mining to identify patterns of fraudulent transactions, such as multiple purchases from the same merchant within a short period of time.

We can then use this information can to flag potentially fraudulent activity and take preventative measures to protect customers.

Social network analysis

Various companies use association rule mining to identify patterns in social media data that can inform the analysis of social networks.

For example, an analysis of Twitter data might reveal that users who tweet about a particular topic are also likely to tweet about other related topics, which could inform the identification of groups or communities within the network.

Recommendation systems

Association rule mining can be used to suggest items that a customer might be interested in based on their past purchases or browsing history. For example, a music streaming service might use association rule mining to recommend new artists or albums to a user based on their listening history.

Association Rule Mining Algorithms

There are several algorithms used for association rule mining. Some common ones are:

Apriori algorithm

The Apriori algorithm is one of the most widely used algorithms for association rule mining. It works by first identifying the frequent itemsets in the dataset (itemsets that appear in a certain number of transactions). It then uses these frequent itemsets to generate association rules, which are statements of the form "if item A is purchased, then item B is also likely to be purchased." The Apriori algorithm uses a bottom-up approach, starting with individual items and gradually building up to more complex itemsets.

FP-Growth algorithm

The FP-Growth (Frequent Pattern Growth) algorithm is another popular algorithm for association rule mining. It works by constructing a tree-like structure called a FP-tree, which encodes the frequent itemsets in the dataset. The FP-tree is then used to generate association rules in a similar manner to the Apriori algorithm. The FP-Growth algorithm is generally faster than the Apriori algorithm, especially for large datasets.

ECLAT algorithm

The ECLAT (Equivalence Class Clustering and bottom-up Lattice Traversal) algorithm is a variation of the Apriori algorithm that uses a top-down approach rather than a bottom-up approach. It works by dividing the items into equivalence classes based on their support (the number of transactions in which they appear). The association rules are then generated by combining these equivalence classes in a lattice-like structure. It is a more efficient and scalable version of the Apriori algorithm.

Apriori Algorithm

The apriori algorithm has become one of the most widely used algorithms for frequent itemset mining and association rule learning. It has been applied to a variety of applications, including market basket analysis, recommendation systems, and fraud detection, and has inspired the development of many other algorithms for similar tasks.

Algorithm Details

The apriori algorithm starts by setting the minimum support threshold. This is the minimum number of times an item must occur in the database in order for it to be considered a frequent itemset. The algorithm then filters out any candidate itemsets that do not meet the minimum support threshold.

The algorithm then generates a list of all possible combinations of frequent itemsets and counts the number of times each combination appears in the database. The algorithm then generates a list of association rules based on the frequent itemset combinations.

An association rule is a statement of the form "if item A is present in a transaction, then item B is also likely to be present". The strength of the association is measured using the confidence of the rule, which is the probability that item B is present given that item A is present.

The algorithm then filters out any association rules that do not meet a minimum confidence threshold. These rules are referred to as strong association rules. Finally, the algorithm then returns the list of strong association rules as output.

Apriori uses a "bottom-up" approach, starting with individual items and gradually combining them into larger and larger itemsets as it searches for frequent patterns. It also uses a "delete-relabel" approach to efficiently prune the search space by eliminating infrequent itemsets from consideration.

Pros:

- Apriori is relatively simple and easy to understand.
- It effectively identifies frequent patterns and association rules in large datasets.
- It is efficient at pruning the search space by eliminating infrequent itemsets from consideration, reducing the algorithm's computational complexity.
- It has been widely used and tested in various applications, making it a well-established and reliable algorithm.

Cons:

- Apriori is not suitable for very large datasets, as the computational complexity of the algorithm increases exponentially with the size of the dataset.
- It may not be as effective at identifying patterns in datasets with many rare items or infrequent transactions.
- It is sensitive to the minimum support and minimum confidence thresholds, which can affect the quality of the results.
- It may be prone to generating a large number of association rules, which can make it difficult to interpret the results.

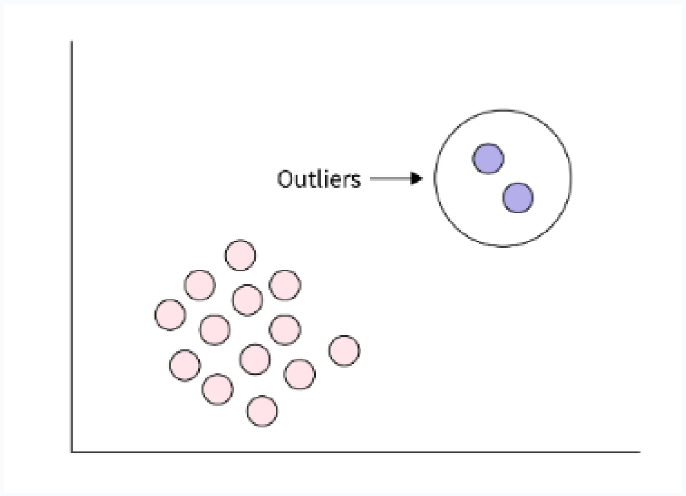
Outlier Analysis in Data Mining

Outlier analysis in data mining involves identifying and analyzing data points significantly different or deviating from the rest of the dataset. Outliers can be caused by various factors, such as data entry errors, unexpected events, etc., and their detection can lead to valuable insights and improve the accuracy of models. A wide range of techniques can be used for outlier analysis in data mining, such as statistical methods, clustering algorithms, and machine learning models.

What is Outlier Analysis in Data Mining

In statistics, an outlier is an observation or data point that significantly differs from other observations in the dataset. Measurement errors, data entry errors, or legitimate deviations in the data can cause outliers.

Outlier analysis in data mining is the process of identifying and examining data points that significantly differ from the rest of the dataset. An outlier can be defined as a data point that deviates significantly from the normal pattern or behavior of the data. Various factors, such as measurement errors, unexpected events, data processing errors, etc., can cause these outliers. For example, outliers are represented as red dots in the figure below, and you can see that they deviate significantly from the rest of the data points. Outliers are also often referred to as anomalies, aberrations, or irregularities.



Why is outlier analysis important?

Outlier analysis is important because it can help identify anomalous data points that can affect the overall analysis and interpretation of the data. By detecting and handling outliers appropriately, data scientists can improve the accuracy and reliability of their results.

Outliers vs. Noise

Outliers differ from noise. In data mining, noise refers to random variations or errors in the data that have no significant meaning or pattern. Noise can arise from various sources, such as measurement errors or data collection methods, and it can negatively affect the accuracy and reliability of data analysis. On the other hand, outliers can provide valuable insights and may need to be studied further, but they can also skew statistical analyses or predictive models if not handled properly.

Overall, the main difference between outliers and noise is that outliers are significant and potentially informative, while noise is insignificant and can be detrimental to data analysis.

Benefits of Outlier Analysis in Data Mining

Outlier analysis in data mining can provide several benefits, as mentioned below -

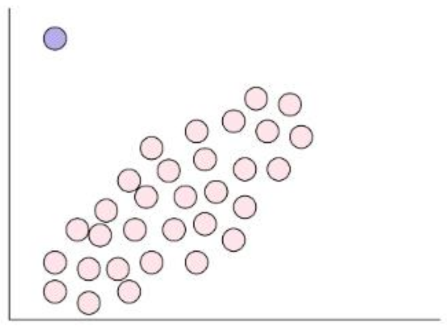
- **Improved accuracy of data analysis** - Outliers can skew the results of statistical analyses or predictive models, leading to inaccurate or misleading conclusions. Detecting and removing outliers can improve the accuracy and reliability of data analysis.

- **Identification of data quality issues** - Outliers can be caused by data collection, processing, or measurement errors, which can indicate data quality issues. Outlier analysis in data mining can help identify and correct these issues to improve data quality.
- **Detection of unusual events or patterns** - Outliers can represent unusual events or patterns in the data that may be of interest to the businesses. Studying these outliers can provide valuable insights and lead to discoveries.
- **Better decision-making** - Outlier analysis in data mining can help decision-makers identify and understand the factors affecting their data, leading to better-informed decisions.
- **Improved model performance** - Outliers can negatively affect the performance of predictive models. Removing outliers or developing models that can handle them appropriately can improve model performance.

Types of Outliers in Data Mining

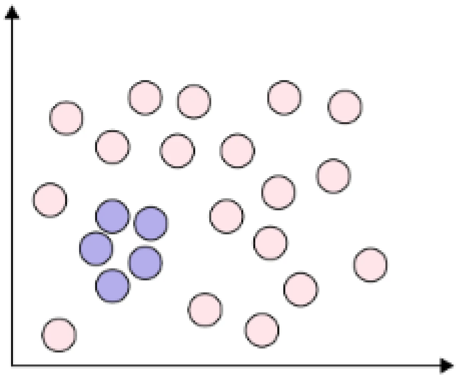
Global (Point) Outliers

These are data points that are significantly different from the rest of the dataset in a global sense. Global outliers are typically detected using statistical methods focusing on the entire dataset's extreme values. For example, if we have a dataset of heights for a group of people, and one person is 7 feet tall while the rest of the heights range between 5 and 6 feet, the height of 7 feet would be a global outlier. An example of a global outlier is also shown below



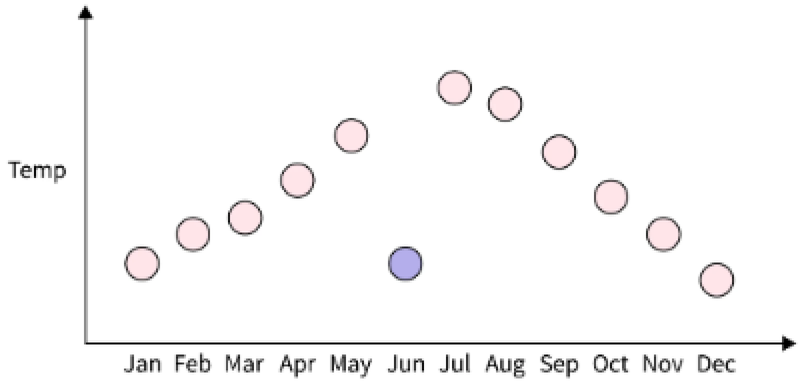
Collective Outliers

These are groups of data points that are significantly different from the rest of the dataset when considered together. Collective outliers are typically detected using clustering algorithms or other methods that group similar data points. For example, suppose we have a dataset of customer transactions, and a group of customers consistently makes purchases that are significantly larger than the rest of the customers. In that case, this group of customers could be considered a collective outlier. Similarly, in an intrusion detection system, the transmission of a DOS packet from one PC to another PC can be considered normal behavior, but if DOS packets are transmitted to many PCs at the same time, it would be considered as collective outliers.



Contextual (Conditional) Outliers

These data points significantly differ from the rest of the dataset in a specific context. Contextual outliers are typically detected using domain knowledge or contextual information relevant to the dataset. For example, if a city is recording 40-degree Celsius temperature, it may be considered normal in the summer and a contextual outlier in the winter. An example of a contextual outlier is shown below -



How and When to Do Outlier Analysis in Data Mining?

Outlier analysis is an important step in data mining as it helps identify and deal with anomalies in the data. Here are some steps involved in the outlier analysis -

- **Identify the data type** - Before performing outlier analysis, it is important to identify the data type being analyzed, as this can impact the choice of outlier detection methods. For example, if the data is continuous, statistical methods such as z-scores can be used, while for categorical data, methods such as the chi-squared test can be used.
- **Understand the context** - It is important to understand the context in which the data was collected, as this can impact what constitutes an outlier. For example, a temperature reading of 40°C might be normal for one location but an outlier for another.

- **Choose appropriate methods** - Once the data type and context have been identified, appropriate outlier detection methods can be chosen. This could include statistical methods, machine learning algorithms such as clustering, or a combination of both.
- **Evaluate and interpret results** - After performing the outlier analysis, evaluating and interpreting the results is important. This involves understanding the outliers detected, determining whether they are genuine anomalies or errors in the data, and deciding on the appropriate actions. This could include removing outliers from the dataset, investigating further to understand the cause of the outliers, or simply noting the presence of outliers without taking further action.

In terms of when to do outlier analysis, it is typically performed as part of the data preprocessing phase before any modeling or analysis is carried out. Outlier analysis can be especially important when working with large datasets or complex data containing many different types of outliers. It is also important to re-evaluate outlier analysis periodically, as new data may reveal previously undetected outliers.

Applications of Outlier Analysis

Outlier analysis has many applications in various fields, as mentioned below -

- **Finance** - In finance, outlier analysis is used to identify abnormal fluctuations in stock prices or financial transactions, which can indicate fraud or insider trading.
- **Healthcare** - Outlier analysis is used in healthcare to identify patients with rare or unusual medical conditions or to detect abnormal patterns in medical data that can help diagnose diseases.
- **Manufacturing** - In manufacturing, outlier analysis is used to identify defective products or equipment producing out-of-specification results, which can affect the quality of the final product.
- **Marketing** - Outlier analysis is used to identify customers with high or low purchasing habits, which can help businesses create targeted marketing campaigns and promotions.
- **Environmental science** - Outlier analysis is used in environmental science to identify extreme weather events or natural disasters, which can help predict and mitigate the impact of these events on human populations and ecosystems.
- **Cybersecurity** - Outlier analysis is used in cybersecurity to detect abnormal network behavior or suspicious activity, indicating cyberattacks or data breaches.

Generalized Sequential Pattern (GSP) Mining

GSP is a very important algorithm in data mining. It is used in sequence mining from large databases. Almost all sequence mining algorithms are basically based on a prior algorithm. GSP uses a level-wise paradigm for finding all the sequence patterns in the data. It starts with finding the frequent items of size one and then passes that as input to the next iteration of the GSP algorithm. The database is passed multiple times to this algorithm. In each iteration, GSP removes all the non-frequent itemsets. This is done based on a threshold frequency which is called support. Only those itemsets are kept whose frequency is greater than the support count. After the first pass, GSP finds all the frequent sequences of length-1 which are called 1-sequences. This makes the input to the next pass, it is the candidate for 2-sequences. At the end of this pass, GSP generates all frequent 2-sequences, which makes the input for candidate 3-sequences. The algorithm is recursively called until no more frequent itemsets are found.

Basic of Sequential Pattern (GSP) Mining:

- **Sequence:** A sequence is formally defined as the ordered set of items {s1, s2, s3, …, sn}. As the name suggests, it is the sequence of items occurring together. It can be considered as a transaction or purchased items together in a basket.
- **Subsequence:** The subset of the sequence is called a subsequence. Suppose {a, b, g, q, y, e, c} is a sequence. The subsequence of this can be {a, b, c} or {y, e}. Observe that the subsequence is not necessarily consecutive items of the sequence. From the sequences of databases, subsequences are found from which the generalized sequence patterns are found at the end.
- **Sequence pattern:** A sub-sequence is called a pattern when it is found in multiple sequences. The goal of the GSP algorithm is to mine the sequence patterns from the large database. The database consists of the sequences. When a subsequence has a frequency equal to more than the "support" value. For example: the pattern <a, b> is a sequence pattern mined from sequences {b, x, c, a}, {a, b, q}, and {a, u, b}.

Sequential Pattern (GSP) Mining uses:

Sequential pattern mining, also known as GSP (Generalized Sequential Pattern) mining, is a technique used to identify patterns in sequential data. The goal of GSP mining is to discover patterns in data that occur over time, such as customer buying habits, website navigation patterns, or sensor data.

Some of the main uses of GSP mining include:

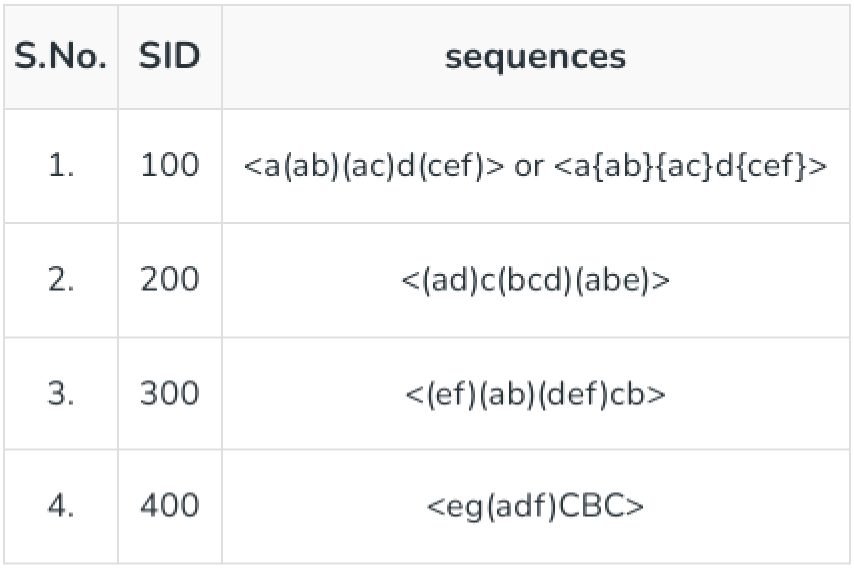
Market basket analysis: GSP mining can be used to analyze customer buying habits and identify products that are frequently purchased together. This can help businesses to optimize their product placement and marketing strategies.

1. **Fraud detection:** GSP mining can be used to identify patterns of behavior that are indicative of fraud, such as unusual patterns of transactions or access to sensitive data.
2. **Website navigation:** GSP mining can be used to analyze website navigation patterns, such as the sequence of pages visited by users, and identify areas of the website that are frequently accessed or ignored.
3. **Sensor data analysis**: GSP mining can be used to analyze sensor data, such as data from IoT devices, and identify patterns in the data that are indicative of certain conditions or states.
4. **Social media analysis**: GSP mining can be used to analyze social media data, such as posts and comments, and identify patterns in the data that indicate trends, sentiment, or other insights.
5. **Medical data analysis**: GSP mining can be used to analyze medical data, such as patient records, and identify patterns in the data that are indicative of certain health conditions or trends.

Methods for Sequential Pattern Mining:

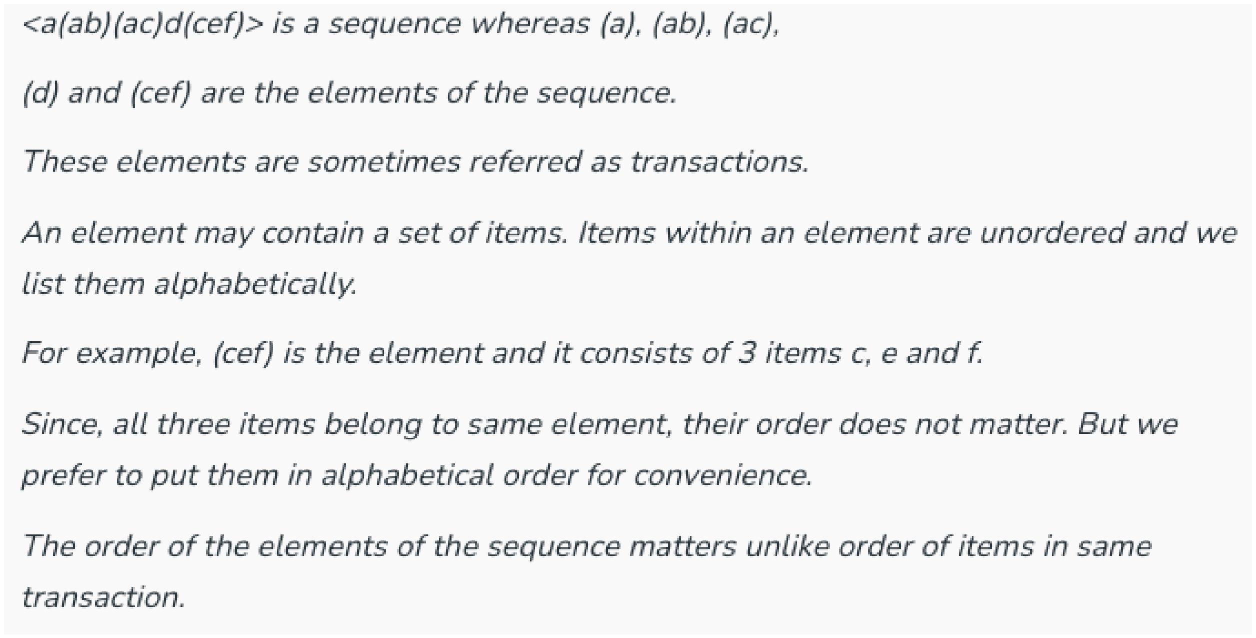
- Apriori-based Approaches
    - GSP
    - SPADE
- Pattern-Growth-based Approaches
    - FreeSpan
    - PrefixSpan

Sequence Database: A database that consists of ordered elements or events is called a sequence database. Example of a sequence database:



S.No.	SID	sequences
1.	100	<a(ab)(ac)d(cef)> or <a{ab}{ac}d{cef}>
2.	200	<(ad)c(bcd)(abe)>
3.	300	<(ef)(ab)(def)cb>
4.	400	<eg(adf)CBC>

Transaction: The sequence consists of many elements which are called transactions.


<a(ab)(ac)d(cef)> is a sequence whereas (a), (ab), (ac),

(d) and (cef) are the elements of the sequence.

These elements are sometimes referred as transactions.

An element may contain a set of items. Items within an element are unordered and we list them alphabetically.

For example, (cef) is the element and it consists of 3 items c, e and f.

Since, all three items belong to same element, their order does not matter. But we prefer to put them in alphabetical order for convenience.

The order of the elements of the sequence matters unlike order of items in same transaction.

k-length Sequence:
The number of items involved in the sequence is denoted by K. A sequence of 2 items is called a 2-len sequence. While finding the 2-length candidate sequence this term comes into use. Example of 2-length sequence is: {ab}, {(ab)}, {bc} and {(bc)}.

- {bc} denotes a 2-length sequence where b and c are two different transactions. This can also be written as {(b)(c)}
- {(bc)} denotes a 2-length sequence where b and c are the items belonging to the same transaction, therefore enclosed in the same parenthesis. This can also be written as {(cb)}, because the order of items in the same transaction does not matter.

Support in k-length Sequence:
Support means the frequency. The number of occurrences of a given k-length sequence in the sequence database is known as the support. While finding the support the order is taken care.

Illustration:
Suppose we have 2 sequences in the database.

s1: <a(bc)b(cd)>

s2: <b(ab)abc(de)>

We need to find the support of {ab} and {(bc)}

Finding support of {ab}:
This is present in first sequence.

s1: <a(bc)b(cd)>

Since, a and b belong to different elements, their order matters.

In second sequence {ab} is not found but {ba} is present.

s2: <b(ab)abc(de)> Thus we don't consider this.

Hence, support of {ab} is 1.

Finding support of {bc}:
Since, b and c are present in same element, their order does not matter.

s1: <a(bc)b(cd)>, first occurrence.

s2: <b(ab)abc(de)>, it seems correct, but is not. b and c are present in different elements here. So, we don't consider it.

Hence, support of {(bc)} is 1.

How to join L1 and L1 to give C2?
L1 is the final 1-length sequence after pruning. After pruning all the entries left in the set have supported greater than the threshold.

Case 1: Join {ab} and {ac}
s1: {ab}, s2: {ac}

After removing a from s1 and c from s2.

s1'={b}, s2'={a}

s1' and s2' are not same, so s1 and s2 can't be joined.

Case 2: Join {ab} and {be}
s1: {ab}, s2: {be}

After removing a from s1 and e from s2.

s1'={b}, s2'={b}

s1' and s2' are exactly same, so s1 and s2 be joined.

s1 + s2 = {abe}

Case 3: Join {(ab)} and {be}
s1: {(ab)}, s2: {be}

After removing a from s1 and e from s2.

s1'={(b)}, s2'={(b)}

s1' and s2' are exactly same, so s1 and s2 be joined.

s1 + s2 = {(ab)e}

s1 and s2 are joined in such a way that items belong to correct elements or transactions.

Pruning Phase: While building Ck (candidate set of k-length), we delete a candidate sequence that has a contiguous (k-1) subsequence whose support count is less than the minimum support (threshold). Also, delete a candidate sequence that has any subsequence without minimum support.
{abg} is a candidate sequence of C3.

{abg} is a candidate sequence of C3.

To check if {abg} is proper candidate or not, without checking its support, we check the support of its subsets.

Because subsets of 3-length sequence will be 1 and 2 length sequences. We build the candidate sets increment like 1-length, 2-length and so on.

Subsets of {abg} are: {ab], {bg} and {ag}

Check support of all three subsets. If any of them have support less than minimum support then delete the sequence {abg} from the set C3 otherwise keep it.

Challenges in Generalized Sequential Pattern Data Mining

The database is passed many times to the algorithm recursively. The computational efforts are more to mine the frequent pattern. When the sequence database is very large and patterns to be mined are long then GSP encounters the problem in doing so effectively.

Difference between Spatial and Temporal Data Mining

1. Spatial Data Mining : Spatial data mining is the process of discovering interesting and previously unknown, but potentially useful patterns from spatial databases. In spatial data mining analyst use geographical or spatial information to produce business intelligence or other results. Challenges involved in spatial data mining include identifying patterns or finding objects that are relevant to research project.
2. Temporal Data Mining : Temporal data refers to the extraction of implicit, non-trivial and potentially useful abstract information from large collection of temporal data. It is concerned with the analysis of temporal data and for finding temporal patterns and regularities in sets of temporal data tasks of temporal data mining are –

- Data Characterization and Comparison
- Cluster Analysis
- Classification
- Association rules

- Prediction and Trend Analysis
- Pattern Analysis

Difference between Spatial and Temporal Data Mining

SNO.	Spatial data mining	Temporal data mining
1.	It requires space.	It requires time.
2.	Spatial mining is the extraction of knowledge/spatial relationship and interesting measures that are not explicitly stored in spatial database.	Temporal mining is the extraction of knowledge about occurrence of an event whether they follow Cyclic , Random ,Seasonal variations etc.
3.	It deals with spatial (location , Geo-referenced) data.	It deals with implicit or explicit Temporal content , from large quantities of data.
4.	Spatial databases reverses spatial objects derived by spatial data. types and spatial association among such objects.	Temporal data mining comprises the subject as well as its utilization in modification of fields.
5.	It includes finding characteristic rules, discriminant rules, association rules and evaluation rules etc.	It aims at mining new and unknown knowledge, which takes into account the temporal aspects of data.
6.	It is the method of identifying unusual and unexplored data but useful models from spatial databases.	It deals with useful knowledge from temporal data.
7.	Examples – Determining hotspots , Unusual locations.	Examples – An association rule which looks like – “Any Person who buys a car also buys steering lock”. By temporal aspect this rule would be – ” Any person who buys a car also buys a steering lock after that “.

Web Mining

Web Mining is the process of Data Mining techniques to automatically discover and extract information from Web documents and services. The main purpose of web mining is to discover useful information from the World Wide Web and its usage patterns.

What is Data Mining?

Web mining is the best type of practice for sifting through the vast amount of data in the system that is available on the World Wide Web to find and extract pertinent information as per requirements. One unique feature of web mining is its ability to deliver a wide range of required data types in the actual process. There are various elements of the web that lead to diverse methods for the actual mining process. For example, web pages are made up of text; they are connected by hyperlinks in the system or process; and web server logs allow for the monitoring of user behavior to simplify all the required systems. Combining all the required methods from data mining, machine learning, artificial intelligence, statistics, and information retrieval, web mining is an interdisciplinary field for the overall system. Analyzing user behavior and website traffic is the one basic type or example of web mining.

Applications of Web Mining

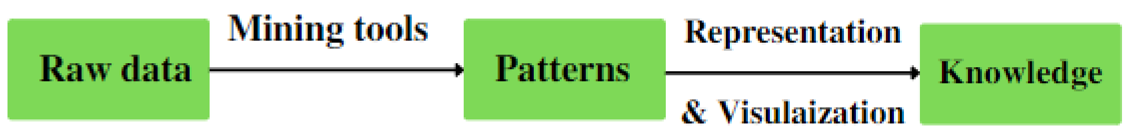
Web mining is the process of discovering patterns, structures, and relationships in web data. It involves using data mining techniques to analyze web data and extract valuable insights. The applications of web mining are wide-ranging and include:

- **Personalized marketing**:Web mining can be used to analyze customer behavior on websites and social media platforms. This information can be used to create personalized marketing campaigns that target customers based on their interests and preferences.
- **E-commerce:** Web mining can be used to analyze customer behavior on e-commerce websites. This information can be used to improve the user experience and increase sales by recommending products based on customer preferences.
- **Search engine optimization:** Web mining can be used to analyze search engine queries and search engine results pages (SERPs). This information can be used to improve the visibility of websites in search engine results and increase traffic to the website.
- **Fraud detection:** Web mining can be used to detect fraudulent activity on websites. This information can be used to prevent financial fraud, identity theft, and other types of online fraud.
- **Sentiment analysis:** Web mining can be used to analyze social media data and extract sentiment from posts, comments, and reviews. This information can be used to understand customer sentiment towards products and services and make informed business decisions.
- **Web content analysis:** Web mining can be used to analyze web content and extract valuable information such as keywords, topics, and themes. This

information can be used to improve the relevance of web content and optimize search engine rankings.

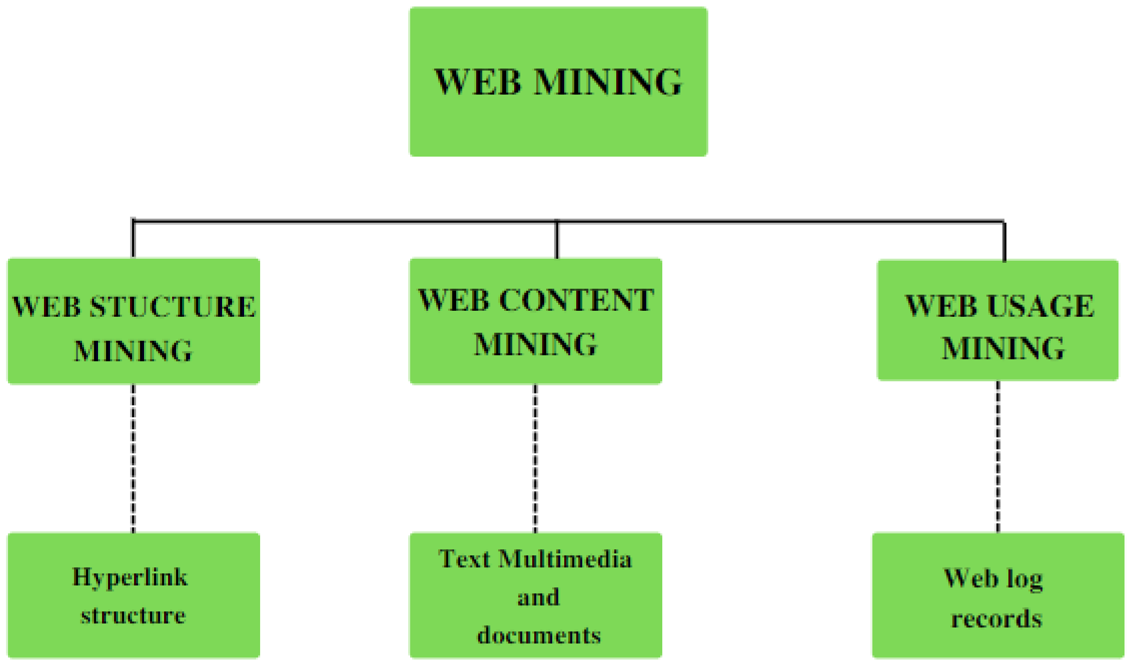
- **Customer service:** Web mining can be used to analyze customer service interactions on websites and social media platforms. This information can be used to improve the quality of customer service and identify areas for improvement.
- **Healthcare:** Web mining can be used to analyze health-related websites and extract valuable information about diseases, treatments, and medications. This information can be used to improve the quality of healthcare and inform medical research.

Process of Web Mining



Web Mining Process

Web mining can be broadly divided into three different types of techniques of mining: Web Content Mining, Web Structure Mining, and Web Usage Mining. These are explained as following below.



- **Web Content Mining:** Web content mining is the application of extracting useful information from the content of the web documents. Web content consist of several types of data – text, image, audio, video etc. Content data is the group of facts that a web page is designed. It can provide effective and interesting patterns about user needs. Text documents are related to text mining, machine learning and natural language processing. This mining is also known as text mining. This type of mining performs scanning and mining of the text, images and groups of web pages according to the content of the input.
- **Web Structure Mining:** Web structure mining is the application of discovering structure information from the web. The structure of the web graph consists of web pages as nodes, and hyperlinks as edges connecting related pages. Structure mining basically shows the structured summary of a particular website. It identifies relationship between web pages linked by information or direct link connection. To determine the connection between two commercial websites, Web structure mining can be very useful.
- **Web Usage Mining:** Web usage mining is the application of identifying or discovering interesting usage patterns from large data sets. And these patterns enable you to understand the user behaviors or something like that. In web usage mining, user access data on the web and collect data in form of logs. So, Web usage mining is also called log mining.

Challenges of Web Mining

- **Complexity of required web pages:** Basically, there is no cohesive framework throughout the site's pages so when compared to conventional text, they are incredibly intricate in the process. The web's digital library contains a vast number of documents in the actual system. There is no set order in which these libraries are typically arranged for the user.
- **Dynamic data source in the internet:** The required online data is updated in real time. For instance, news, weather, fashion, finance, sports, and so forth is not possible to indicate properly.
- **Data relevancy:** It is much believed that a particular person is typically only concerned with a limited percentage of the internet throughout the process, with the remaining portion containing data that may provide unexpected outcomes for the actual requirement and is unfamiliar to the user to verify.
- **Too much large web:** Basically, the web is getting bigger and bigger very quickly in the system. The web seems to be too big for data mining and data warehousing as per requirement.