VIPS
Technical Campus
योग: कर्मसु कौशलम्
IN PURSUIT OF PERFECTION

SCHOOL OF ENGINEERING AND TECHNOLOGY

Unit 3

- **UNIT III:** Introduction of Fuzzy Reasoning and Neural Networks.

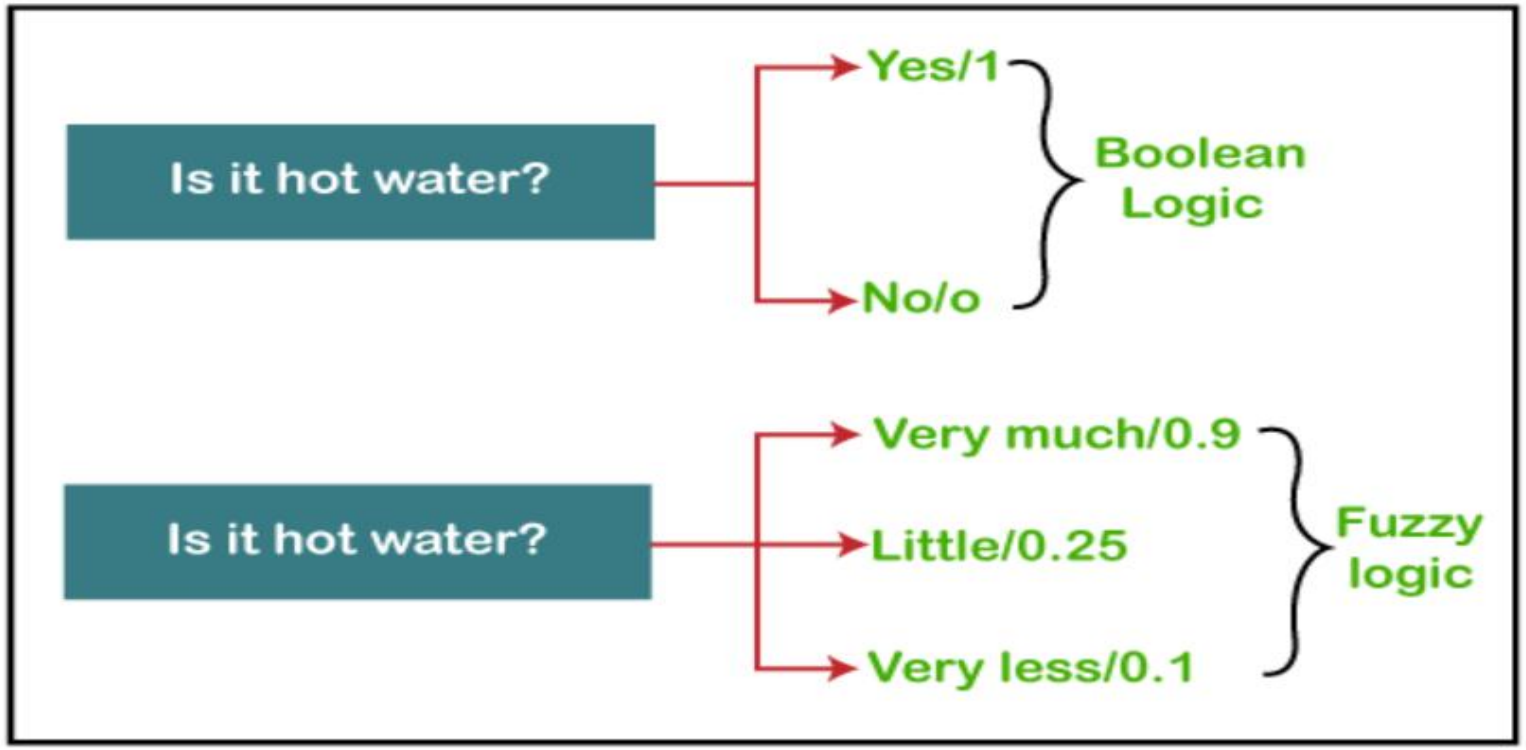
Video Link

- https://youtu.be/f1a6lPPQNgM

Fuzzy Logic

- The 'Fuzzy' word means the things that are not clear or are vague. Sometimes, we cannot decide in real life that the given problem or statement is either true or false. At that time, this concept provides many values between the true and false and gives the flexibility to find the best solution to that problem.

Example of Fuzzy Logic as comparing to Boolean Logic



Fuzzy logic contains the multiple logical values and these values are the truth values of a variable or problem between 0 and 1. This concept was introduced by **Lofti Zadeh** in **1965** based on the **Fuzzy Set Theory**. This concept provides the possibilities which are not given by computers, but similar to the range of possibilities generated by humans.

In the Boolean system, only two possibilities (0 and 1) exist, where 1 denotes the absolute truth value and 0 denotes the absolute false value. But in the fuzzy system, there are multiple possibilities present between the 0 and 1, which are partially false and partially true.

The Fuzzy logic can be implemented in systems such as micro-controllers, workstation-based or large network-based systems for achieving the definite output. It can also be implemented in both hardware or software.

Characteristics of Fuzzy Logic

Characteristics of Fuzzy Logic

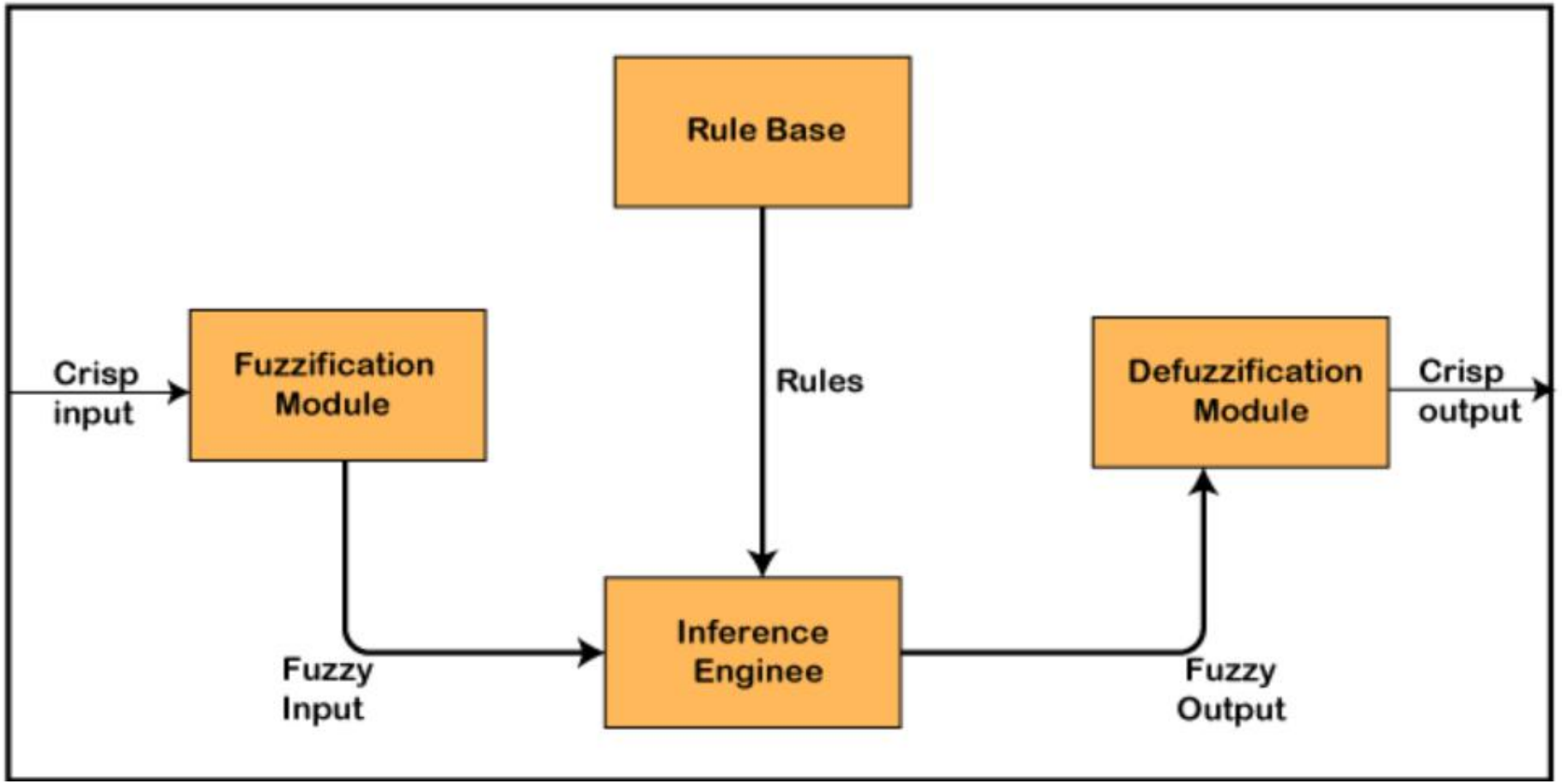
Following are the characteristics of fuzzy logic:

1. This concept is flexible and we can easily understand and implement it.
2. It is used for helping the minimization of the logics created by the human.
3. It is the best method for finding the solution of those problems which are suitable for approximate or uncertain reasoning.
4. It always offers two values, which denote the two possible solutions for a problem and statement.
5. It allows users to build or create the functions which are non-linear of arbitrary complexity.
6. In fuzzy logic, everything is a matter of degree.
7. In the Fuzzy logic, any system which is logical can be easily fuzzified.
8. It is based on natural language processing.
9. It is also used by the quantitative analysts for improving their algorithm's execution.
10. It also allows users to integrate with the programming.

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


Rule Base
Crisp input
Fuzzification Module
Rules
Defuzzification Module
Crisp output
Fuzzy Input
Inference Enginee
Fuzzy Output

1. Rule Base

1. 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.

Fuzzification

Fuzzification is a module or component for transforming the system inputs, i.e., it converts the crisp number into fuzzy steps. 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:

- Large Positive (LP)
- Medium Positive (MP)
- Small (S)
- Medium Negative (MN)
- Large negative (LN)

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.

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.

This function of Membership was introduced in the first papers of fuzzy set by **Zadeh**. For the Fuzzy set B, the membership function for X is defined as: $\mu B:X \rightarrow [0,1]$. 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.

Fuzzy Sets (Table from Rich and Knight)

Table 22.1 *Ages and their memberships*

Age	*Infant*	*Child*	*Adolescent*	*Young*	*Adult*	*Old*
2	1	0	0	1	0	0
4	0.1	0.5	0	1	0	0
10	0	1	0.3	1	0	0
15	0	0.8	1	1	0	0
21	0	0	0.1	1	0.8	0.1
30	0	0	0	0.6	1	0.3
35	0	0	0	0.5	1	0.35
40	0	0	0	0.4	1	0.4
45	0	0	0	0.2	1	0.6
60	0	0	0	0	1	0.8
70	0	0	0	0	1	1

The values in the table indicate memberships to the fuzzy sets – *infant, child, adolescent, young, adult* and *old*. Thus a child of age 4 belongs only 50% to the fuzzy set *child* while when he is 10 years he is a 100% member. Note that membership is different from probabilities. Memberships do not necessarily add up to 1. The entries in the table have been made after a manual evaluation of the different ages.

Fuzzy Terminology

Universe of Discourse (U):

This is defined as the range of all possible values that comprise the input to the fuzzy system.

Fuzzy Set

Any set that empowers its members to have different grades of membership (based on a membership function) in an interval [0,1] is a fuzzy set.

Membership function

The membership function μ_A which forms the basis of a fuzzy set is given by

$$\mu_A: U \rightarrow [0,1]$$

where the closed interval is one that holds real numbers.

Support of a fuzzy set (S_f)

The support S of a fuzzy set f, in a universal crisp set U is that set which contains all elements of the set U that have a non-zero membership value in f. For instance, the support of the fuzzy set *adult* is

$$S_{adult} = \{21.30.35,40.45,60,70\}$$

Depiction of a fuzzy set

A fuzzy set f in a universal crisp set U, is written as

$$f = \mu_1 / s_1 + \mu_2 / s_2 + \mu_3 / s_3 + \ldots + \mu_n / s_n$$

where μ_i is the membership and s_i is the corresponding term in the *support* set of f i.e. S_f.

This is however only a representation and has *no algebraic implication* (the slash and + signs do not have any meaning).

Accordingly,

$$\text{Old} = 0.1/21 + 0.3/30 + 0.35/35 + 0.4/40 + 0.6/45 + 0.8/60 + 1/70$$

Fuzzy Set Operations

Fuzzy Set Operations

- **Union:** The membership function of the union of two fuzzy sets A and B is defined as the maximum of the two individual membership functions. It is equivalent to the Boolean OR operation.

$$\mu_{A \cup B} = \max(\mu_A, \mu_B)$$

- **Intersection:** The membership function of the intersection of two fuzzy sets A and B is defined as the minimum of the two individual membership functions and is equivalent to the Boolean AND operation.

$$\mu_{A \cap B} = \min(\mu_A, \mu_B)$$

- **Complement:** The membership function of the complement of a fuzzy set A is defined as the negation of the specified membership function: $\mu_{\bar{A}}$. This is equivalent to the Boolean NOT operation

$$\mu_{\bar{A}} = \mu_{A \cup B} = (1 - \mu_A)$$

It may be further noted here that the laws of Associativity, Commutativity , Distributivity and De Morgan's laws hold in fuzzy set theory too.

Fuzzy Set Operations: Union

Example:

Let's suppose A is a set which contains following elements:

$$A = \{(X_1, 0.6), (X_2, 0.2), (X_3, 1), (X_4, 0.4)\}$$

And, B is a set which contains following elements:

$$B = \{(X_1, 0.1), (X_2, 0.8), (X_3, 0), (X_4, 0.9)\}$$

then,

$$AUB = \{(X_1, 0.6), (X_2, 0.8), (X_3, 1), (X_4, 0.9)\}$$

Fuzzy Set Operations: Intersection

Example:

Let's suppose A is a set which contains following elements:

$$A = \{(X_1, 0.3), (X_2, 0.7), (X_3, 0.5), (X_4, 0.1)\}$$

And, B is a set which contains following elements:

$$B = \{(X_1, 0.8), (X_2, 0.2), (X_3, 0.4), (X_4, 0.9)\}$$

then,

$$A \cap B = \{(X_1, 0.3), (X_2, 0.2), (X_3, 0.4), (X_4, 0.1)\}$$

Fuzzy Set Operations: Compliment

Example:

Let's suppose A is a set which contains following elements:

$$A = \{(X_1, 0.3), (X_2, 0.8), (X_3, 0.5), (X_4, 0.1)\}$$

then,

$$\bar{A} = \{(X_1, 0.7), (X_2, 0.2), (X_3, 0.5), (X_4, 0.9)\}$$

Difference Between Fuzzy and Classical Set Theory

Classical Set Theory	Fuzzy Set Theory
1. This theory is a class of those sets having sharp boundaries.	1. This theory is a class of those sets having un-sharp boundaries.
2. This set theory is defined by exact boundaries only 0 and 1.	2. This set theory is defined by ambiguous boundaries.
3. In this theory, there is no uncertainty about the boundary's location of a set.	3. In this theory, there always exists uncertainty about the boundary's location of a set.
4. This theory is widely used in the design of digital systems.	4. It is mainly used for fuzzy controllers.

Applications of Fuzzy Logic

Following are the different application areas where the Fuzzy Logic concept is widely used:

1. It is used in **Businesses** for decision-making support system.
2. It is used in **Automative systems** for controlling the traffic and speed, and for improving the efficiency of automatic transmissions. **Automative systems** also use the shift scheduling method for automatic transmissions.
3. This concept is also used in the **Defence** in various areas. Defence mainly uses the Fuzzy logic systems for underwater target recognition and the automatic target recognition of thermal infrared images.
4. It is also widely used in the **Pattern Recognition and Classification** in the form of Fuzzy logic-based recognition and handwriting recognition. It is also used in the searching of fuzzy images.
5. Fuzzy logic systems also used in **Securities**.

Applications of Fuzzy Logic (Contd.)

6. It is also used in **microwave oven** for setting the lunes power and cooking strategy.
7. This technique is also used in the area of **modern control systems** such as expert systems.
8. **Finance** is also another application where this concept is used for predicting the stock market, and for managing the funds.
9. It is also used for controlling the brakes.
10. It is also used in the **industries of chemicals** for controlling the ph, and chemical distillation process.
11. It is also used in the **industries of manufacturing** for the optimization of milk and cheese production.
12. It is also used in the vacuum cleaners, and the timings of washing machines.
13. It is also used in heaters, air conditioners, and humidifiers.

Advantages of Fuzzy Logic

1. The methodology of this concept works similarly as the human reasoning.
2. Any user can easily understand the structure of Fuzzy Logic.
3. It does not need a large memory, because the algorithms can be easily described with fewer data.
4. It is widely used in all fields of life and easily provides effective solutions to the problems which have high complexity.
5. This concept is based on the set theory of mathematics, so that's why it is simple.
6. It allows users for controlling the control machines and consumer products.
7. The development time of fuzzy logic is short as compared to conventional methods.
8. Due to its flexibility, any user can easily add and delete rules in the FLS system.

Disadvantages of Fuzzy Logic

1. The run time of fuzzy logic systems is slow and takes a long time to produce outputs.
2. Users can understand it easily if they are simple.
3. The possibilities produced by the fuzzy logic system are not always accurate.
4. Many researchers give various ways for solving a given statement using this technique which leads to ambiguity.
5. Fuzzy logics are not suitable for those problems that require high accuracy.
6. The systems of a Fuzzy logic need a lot of testing for verification and validation.

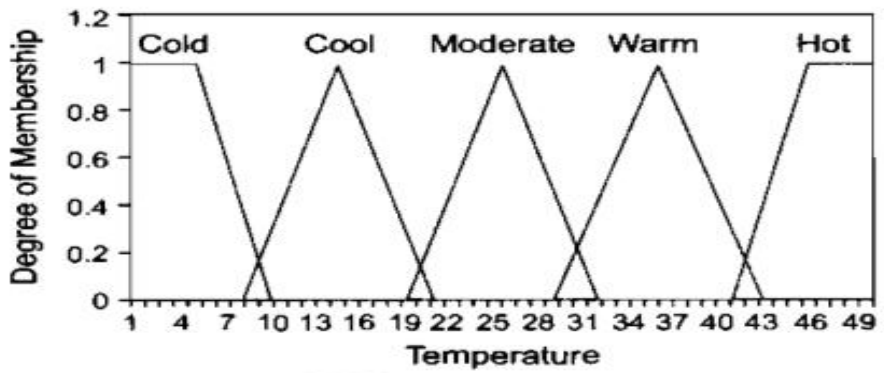
Case Study: Fuzzy Room Cooler

- **Fuzzy Regions**

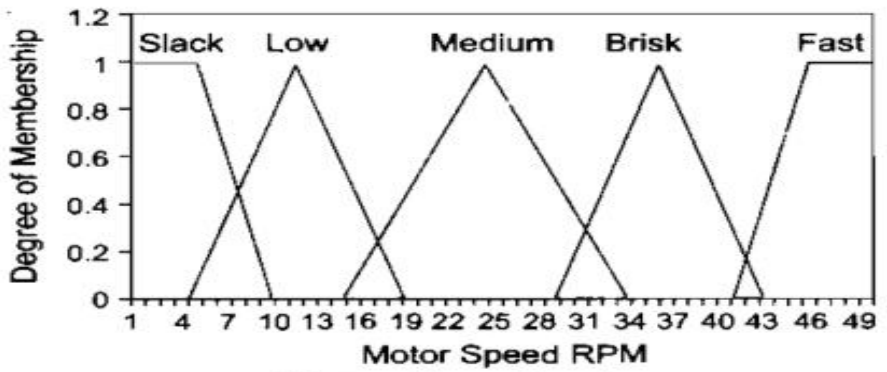
- **Two parameters decide the water flow rate (Temperature and Pressure)**
  - **Fuzzy Terms for Temperature: Cold, Cool, Moderate, Warm and Hot**
  - **Fuzzy Terms for Fan Motor Speed: Slack, Low, Medium, Brisk, Fast.**

Output of the System, which is the flow-rate of the water controlled by the motorized pump, coould also be defined accordingly by yet another set of fuzzy terms: Strong-Negative, Negative, Low-Negative, Low-Positive and High Positive

Fuzzy Profiles



(a) Temperature



(b) Fan Motor Speed

Fig. 22.2

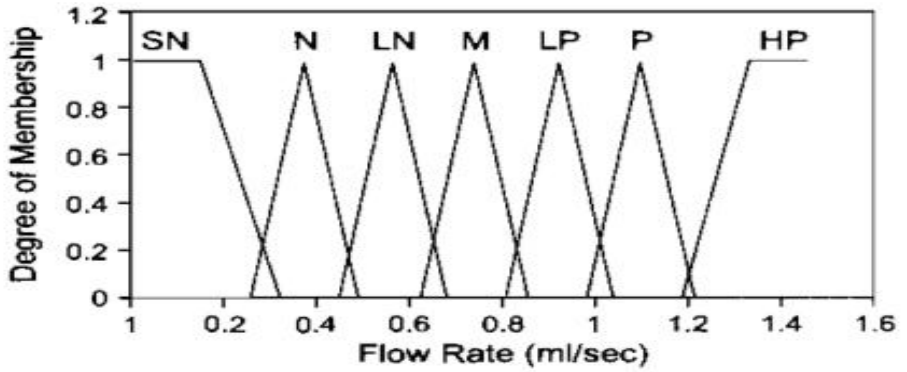


Fig. 22.3 *Water Flow Rate*

Fuzzy Rules

Fuzzy Rules

The fuzzy rules form the triggers of the fuzzy engine. After a study of the system we could write linguistic rules (so akin to natural language) such as –

R1: If temperature is HOT ***and*** fan motor speed is SLACK then flow-rate is HIGH-POSITIVE.
R2: If temperature is HOT ***and*** fan motor speed is LOW then flow-rate is HIGH-POSITIVE
R3: If temperature is HOT ***and*** fan motor speed is MEDIUM then the flow-rate is POSITIVE.
R4: If temperature is HOT ***and*** fan motor speed is BRISK then the flow-rate is HIGH-POSITIVE.
R5: If temperature is WARM ***and*** fan motor speed is MEDIUM then the flow-rate is LOW-POSITIVE,
R6: If temperature is WARM ***and*** fan motor speed is BRISK then the flow-rate is POSITIVE.
R7: If temperature is COOL ***and*** fan motor speed is LOW then flow-rate is NEGATIVE.
R8: If temperature is MODERATE ***and*** fan motor speed is LOW then flow-rate is MEDIUM.

Fuzzification

The fuzzifier forms the heart of the fuzzy engine. Whenever the sensors report the values of temperature and fan speed, they are mapped based on their memberships to the respective fuzzy regions they belong to. For instance if at some instance of time t the temperature is 42 degrees and fan speed is 31 rpm, the corresponding membership values and the associated fuzzy regions are mentioned below

Parameter	*Fuzzy Regions*	*Memberships*
Temperature	warm, hot	0.142, 0.2
Fan speed	medium, brisk	0.25,0.286

From the table, since both temperature and fan speed belong to two regions, it is clear that the rules R3, R4, R5 and R6 are applicable. The rules indicate a conflict. While two of them state that the flow-rate should be POSITIVE, the other two state that it should be LOW-POSITIVE and HIGH-POSITIVE respectively. Though we have resolved the issue of what could be the flow rates, the actual crisp value still eludes us.

Defuzzification

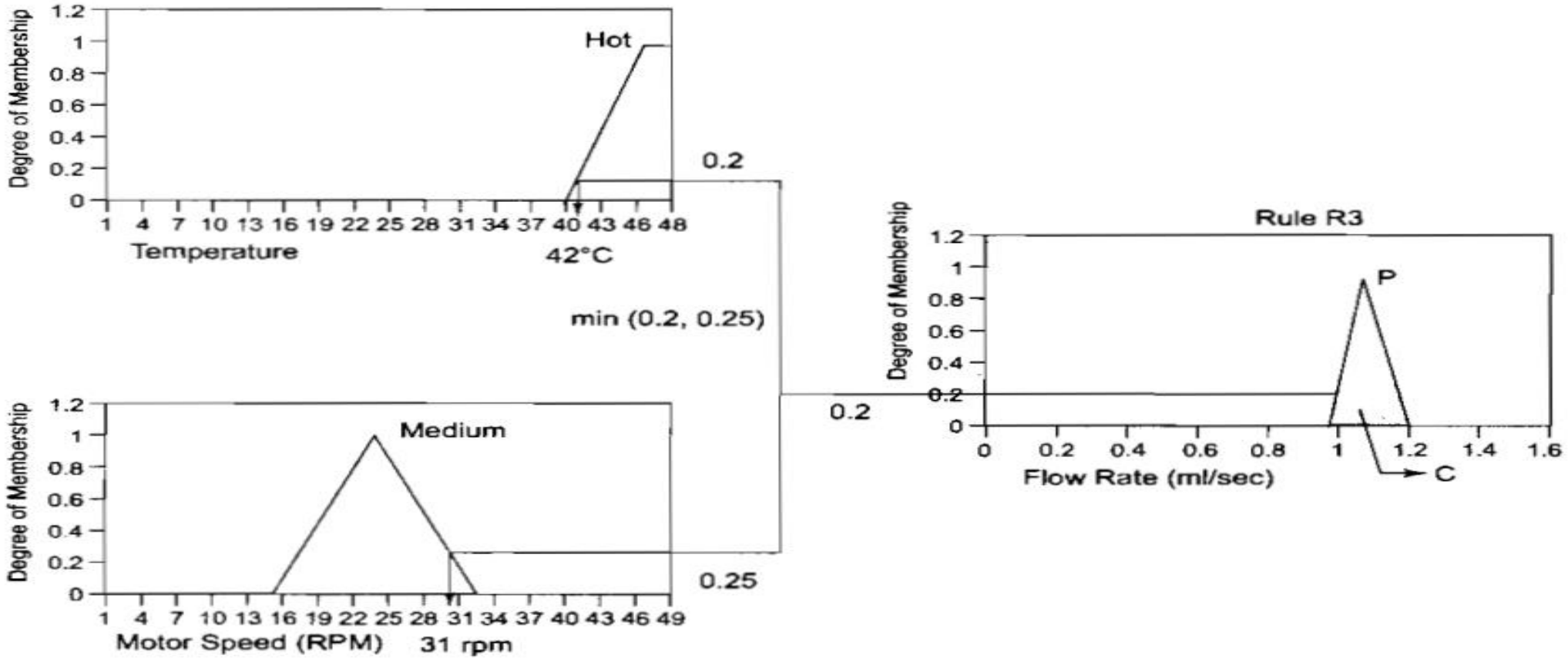
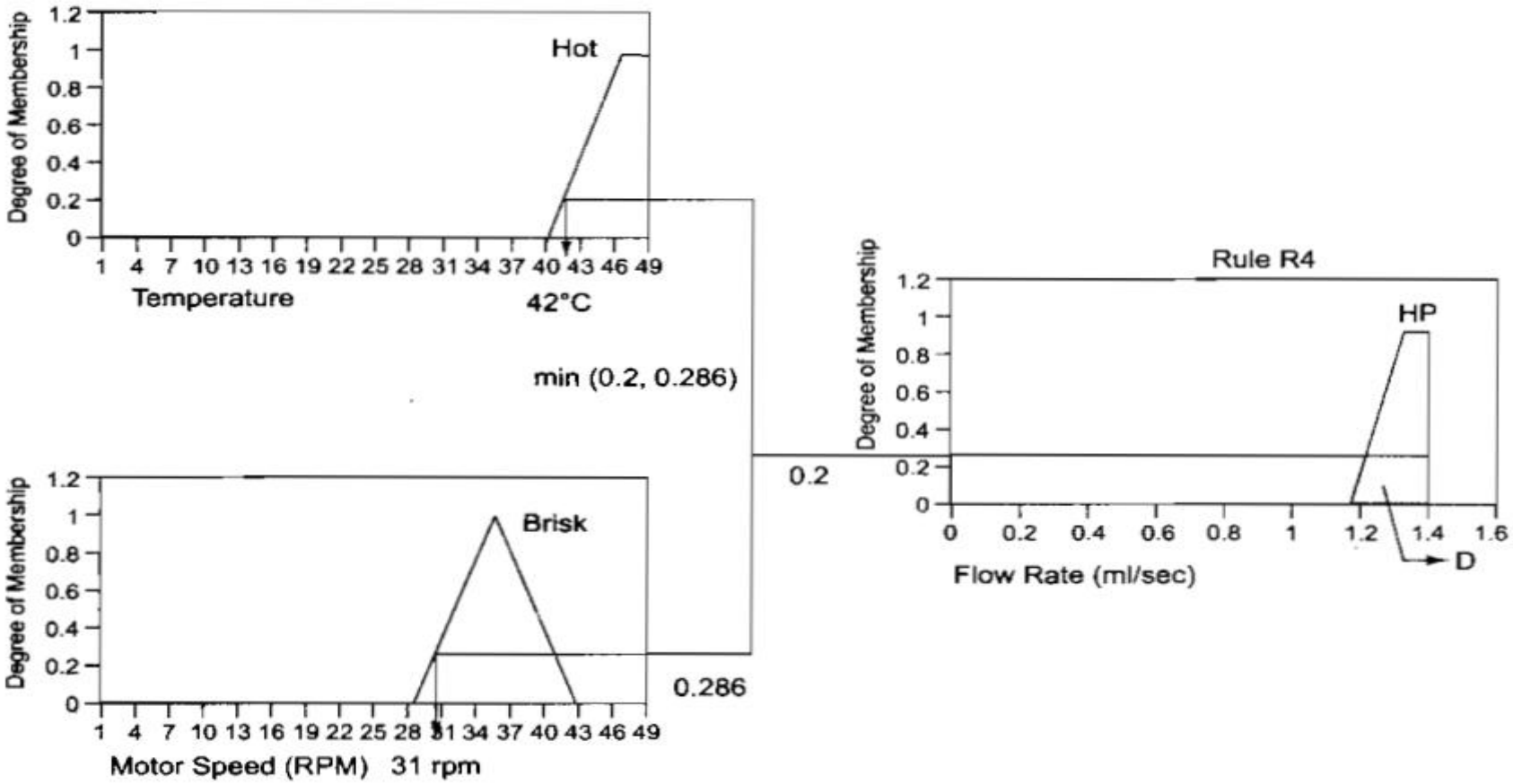


Fig. 22.4 *Defuzzification (contd.)*

Defuzzification (Contd.)



Defuzzification (Contd.)

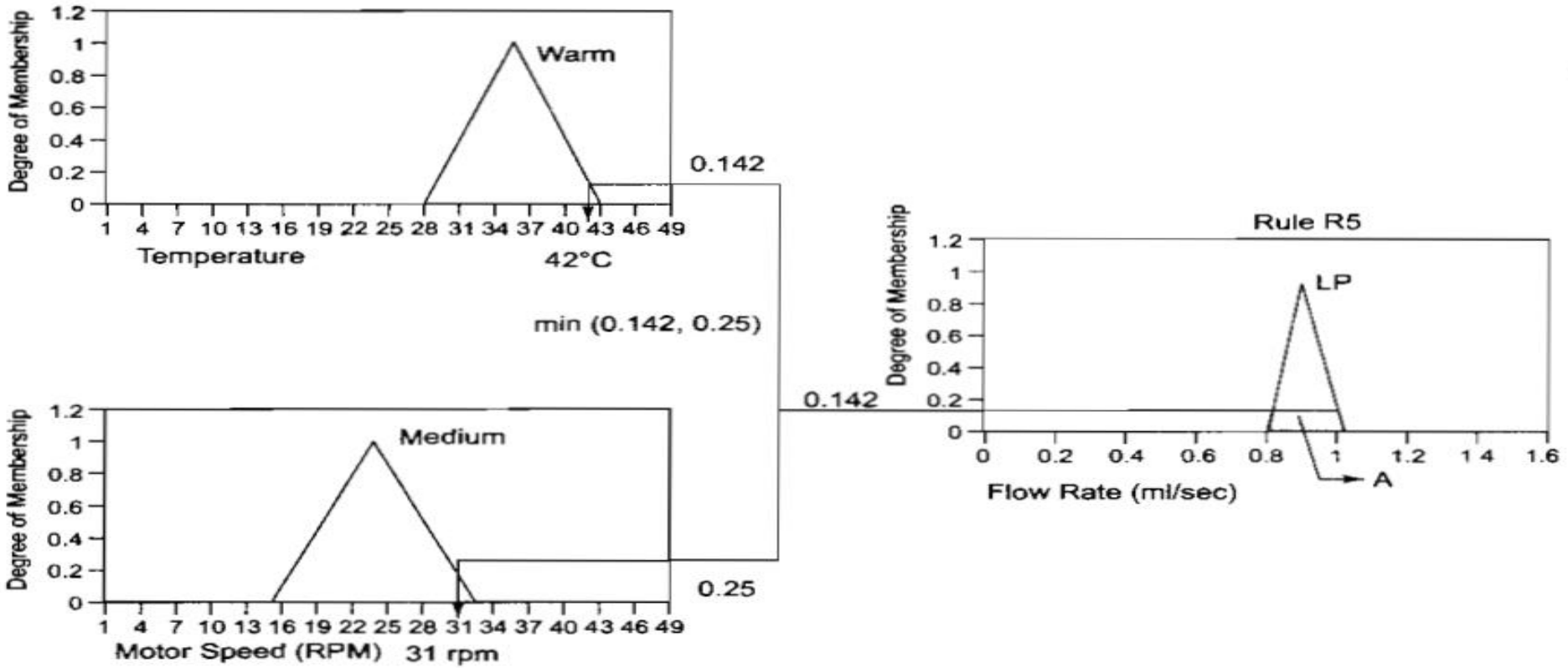
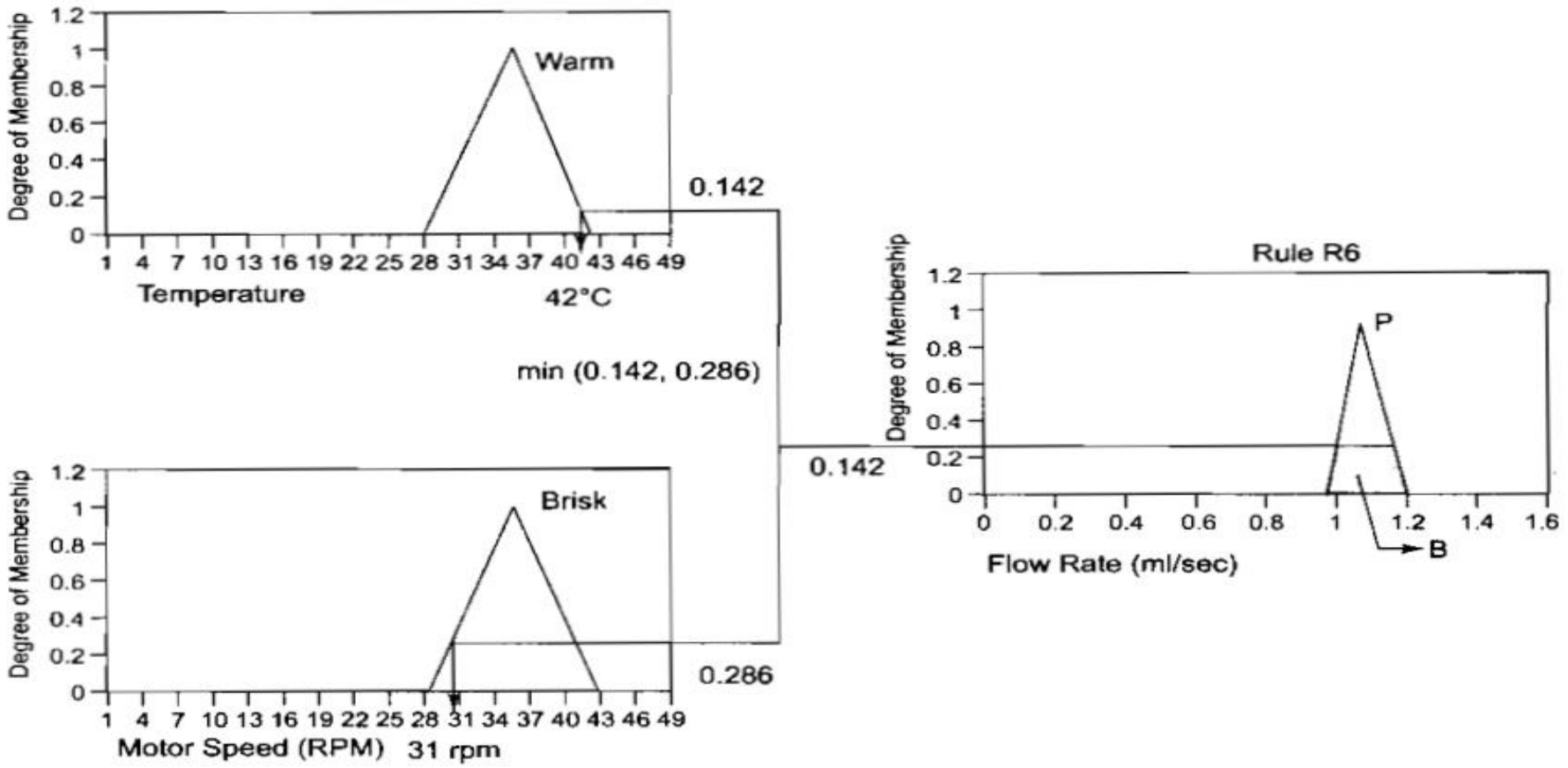


Fig. 22.4 *Defuzzification (contd.)*

Defuzzification (Contd.)



Defuzzification (Contd.)

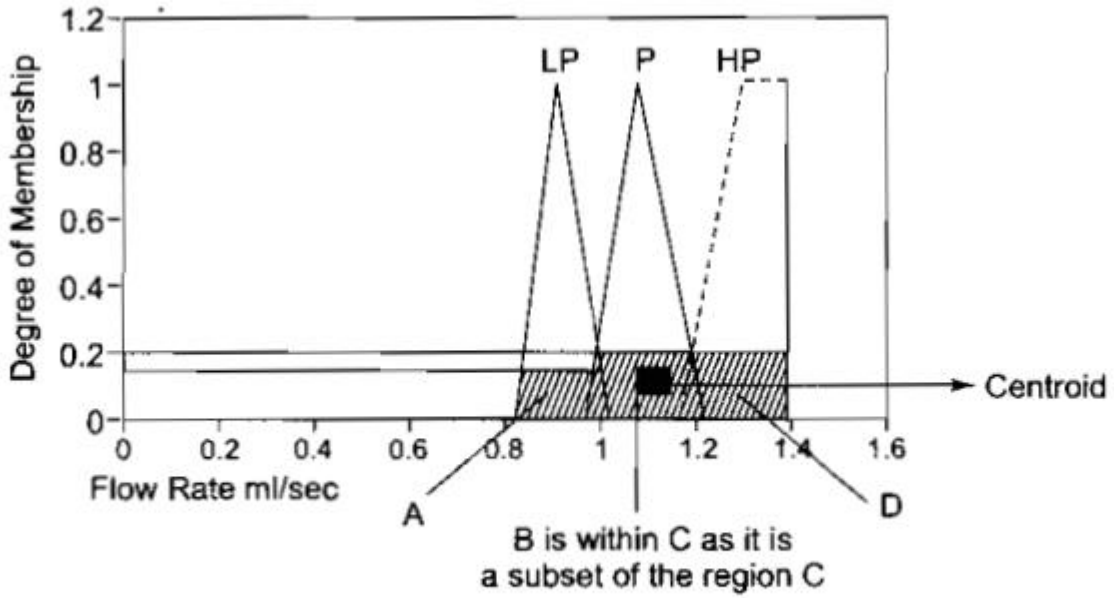


Fig. 22.4 *Defuzzification*

Defuzzifier

This is where we have to demystify these fuzzy terms for the flow rate controller system. In other words, the fuzzy outputs LOW-POSITIVE, POSITIVE and HIGH-POSITIVE are to be converted to a single crisp value which can then be delivered to the final actuator of the pump. This process is called defuzzification. Several methods are used to achieve defuzzification, the most common ones being the Centre of Gravity method and the Composite Maxima method. In both these methods we need to compute the composite region formed by the portions A, B, C and D (See Fig. 22.4) on the output profile. Figure 22.4 shows how this is calculated. In case of parameters whose premises are connected by an *AND*, the minimum of their memberships is first found. This value is used to cut through the profile of the output fuzzy set (done by drawing a horizontal line). This results in a region (area) on the output surface. For cases where an *OR* relates the premises the maximum membership is taken to work out the output surface. All output surfaces are found to obtain the composite output region.

Depending on the application, either the Centre of Gravity or the Composite Maxima of this region (area) is found and treated as the crisp output. The former method works best for control applications such as the one described herein. The crisp output is the desired flow rate (X-coordinate of the Centroid) and the motorized pump is adjusted accordingly based on this value.

Summary

Steps involved in Fuzzy Logic Based Systems

(i) Formulating Fuzzy regions,
(ii) Fuzzy Rules and
(iii) Embedding a Defuzzification procedure.

Fuzzy logic has also been widely applied to non-control applications as well. Take the case of deciding whether a book on Artificial Intelligence belongs to the domain of Computer Science, Psychology or Civil Engineering. In such situations a crisp numerical output obtained by this defuzzification process used earlier may carry no meaning. The composite maxima is generally used for such problems and the truth depends on whether or not the composite maxima has crossed a predetermined threshold. The defuzzification process however may be tuned to suit the satisfiability of the application at hand.

NeuroFuzzy Room Cooler

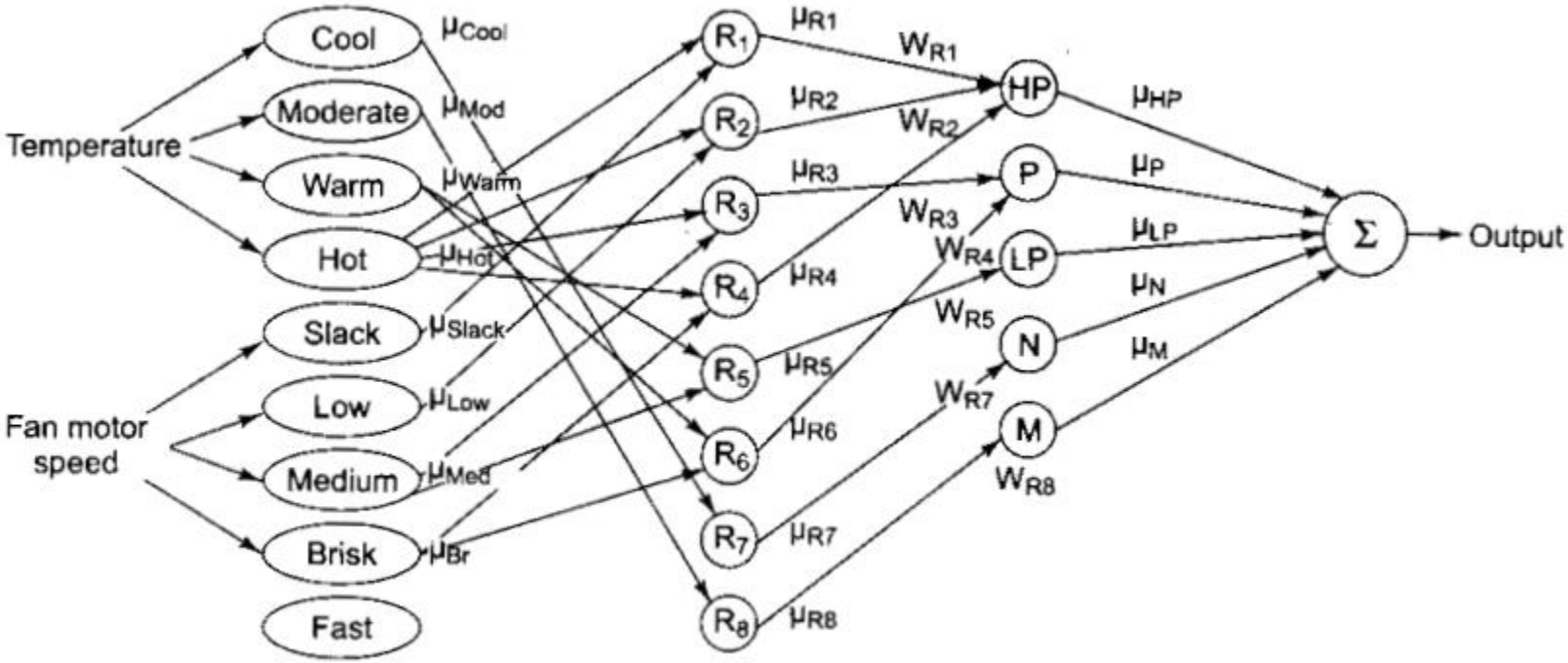


Fig. 22.7 *Neuro Fuzzy Room Cooler*

VIPS Technical Campus
योग: कर्मसु कौशलम्
IN PURSUIT OF PERFECTION

SCHOOL OF ENGINEERING AND TECHNOLOGY

THANK YOU

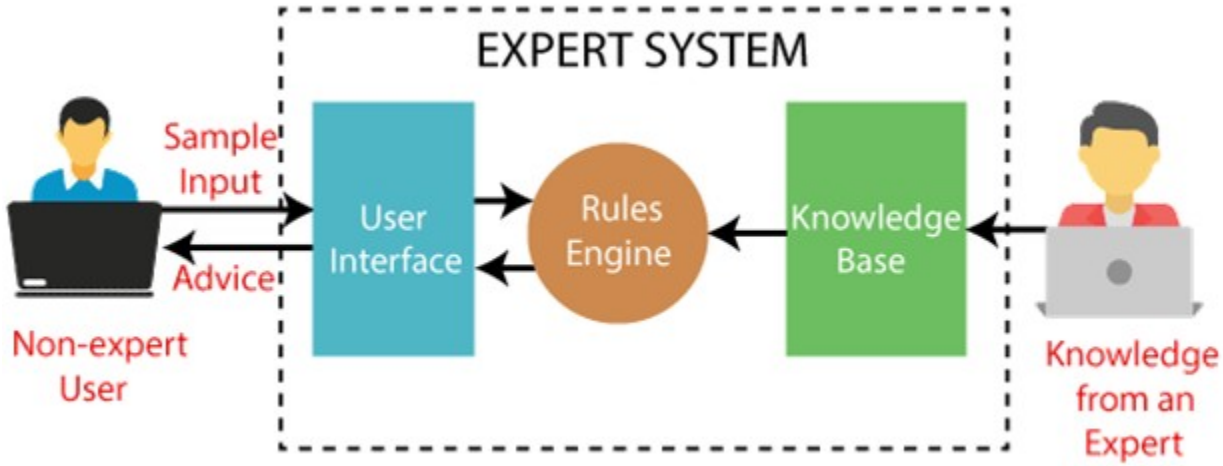
What is an Expert System?

An expert system is a computer program that is designed to solve complex problems and to provide decision-making ability like a human expert. It performs this by extracting knowledge from its knowledge base using the reasoning and inference rules according to the user queries.

The expert system is a part of AI, and the first ES was developed in the year 1970, which was the first successful approach of artificial intelligence. It solves the most complex issue as an expert by extracting the knowledge stored in its knowledge base. The system helps in decision making for complex problems using **both facts and heuristics like a human expert**. It is called so because it contains the expert knowledge of a specific domain and can solve any complex problem of that particular domain. These systems are designed for a specific domain, such as **medicine, science,** etc.

The performance of an expert system is based on the expert's knowledge stored in its knowledge base. The more knowledge stored in the KB, the more that system improves its performance. One of the common examples of an ES is a suggestion of spelling errors while typing in the Google search box.

Below is the block diagram that represents the working of an expert system:



Note: It is important to remember that an expert system is not used to replace the human experts; instead, it is used to assist the human in making a complex decision. These systems do not have human capabilities of thinking and work on the basis of the knowledge base of the particular domain.

Below are some popular examples of the Expert System:

- **DENDRAL:** It was an artificial intelligence project that was made as a chemical analysis expert system. It was used in organic chemistry to detect unknown organic molecules with the help of their mass spectra and knowledge base of chemistry.
- **MYCIN:** It was one of the earliest backward chaining expert systems that was designed to find the bacteria causing infections like bacteraemia and meningitis. It was also used for the recommendation of antibiotics and the diagnosis of blood clotting diseases.
- **PXDES:** It is an expert system that is used to determine the type and level of lung cancer. To determine the disease, it takes a picture from the upper body, which looks like the shadow. This shadow identifies the type and degree of harm.
- **CaDeT:** The CaDet expert system is a diagnostic support system that can detect cancer at early stages.

Characteristics of Expert System

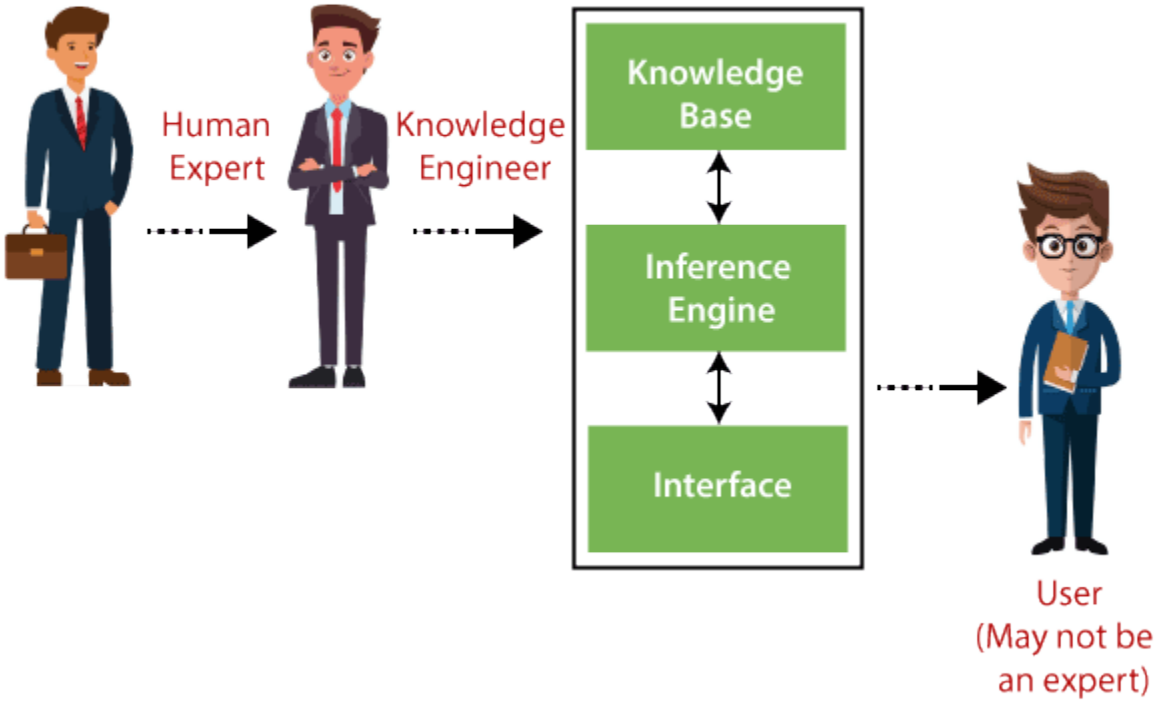
- **High Performance:** The expert system provides high performance for solving any type of complex problem of a specific domain with high efficiency and accuracy.
- **Understandable:** It responds in a way that can be easily understandable by the user. It can take input in human language and provides the output in the same way.
- **Reliable:** It is much reliable for generating an efficient and accurate output.
- **Highly responsive:** ES provides the result for any complex query within a very short period of time.

Components of Expert System

An expert system mainly consists of three components:

- **User Interface**
- **Inference Engine**

- **Knowledge Base**



1. User Interface

With the help of a user interface, the expert system interacts with the user, takes queries as an input in a readable format, and passes it to the inference engine. After getting the response from the inference engine, it displays the output to the user. In other words, **it is an interface that helps a non-expert user to communicate with the expert system to find a solution**.

2. Inference Engine(Rules of Engine)

- The inference engine is known as the brain of the expert system as it is the main processing unit of the system. It applies inference rules to the knowledge base to derive a conclusion or deduce new information. It helps in deriving an error-free solution of queries asked by the user.
- With the help of an inference engine, the system extracts the knowledge from the knowledge base.
- There are two types of inference engine:

- **Deterministic Inference engine:** The conclusions drawn from this type of inference engine are assumed to be true. It is based on **facts** and **rules**.
- **Probabilistic Inference engine:** This type of inference engine contains uncertainty in conclusions, and based on the probability.

Inference engine uses the below modes to derive the solutions:

- **Forward Chaining:** It starts from the known facts and rules, and applies the inference rules to add their conclusion to the known facts.
- **Backward Chaining:** It is a backward reasoning method that starts from the goal and works backward to prove the known facts.

3. Knowledge Base

- The knowledgebase is a type of storage that stores knowledge acquired from the different experts of the particular domain. It is considered as big storage of knowledge. The more the knowledge base, the more precise will be the Expert System.
- It is similar to a database that contains information and rules of a particular domain or subject.
- One can also view the knowledge base as collections of objects and their attributes. Such as a Lion is an object and its attributes are it is a mammal, it is not a domestic animal, etc.

Components of Knowledge Base

- **Factual Knowledge:** The knowledge which is based on facts and accepted by knowledge engineers comes under factual knowledge.
- **Heuristic Knowledge:** This knowledge is based on practice, the ability to guess, evaluation, and experiences.

Knowledge Representation: It is used to formalize the knowledge stored in the knowledge base using the If-else rules.

Knowledge Acquisitions: It is the process of extracting, organizing, and structuring the domain knowledge, specifying the rules to acquire the knowledge from various experts, and store that knowledge into the knowledge base.

Development of Expert System

Here, we will explain the working of an expert system by taking an example of MYCIN ES. Below are some steps to build an MYCIN:

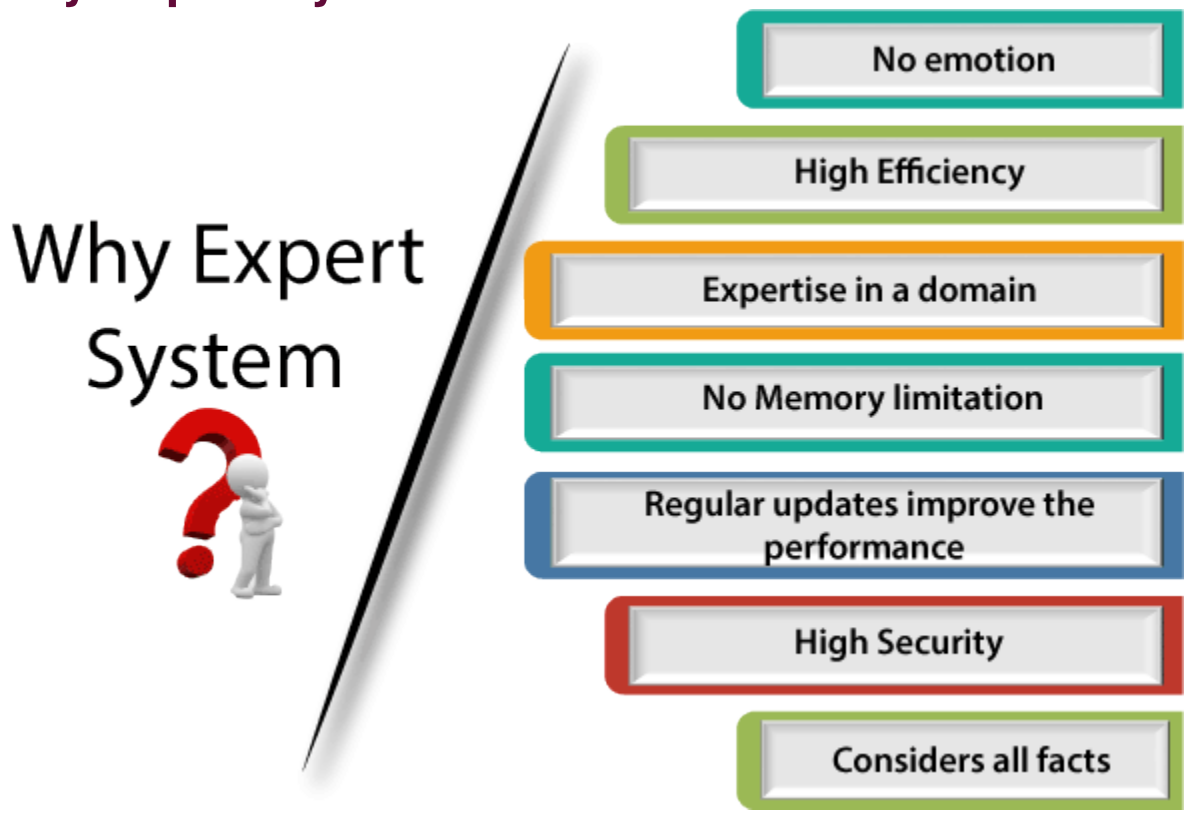
- Firstly, ES should be fed with expert knowledge. In the case of MYCIN, human experts specialized in the medical field of bacterial infection, provide information about the causes, symptoms, and other knowledge in that domain.
- The KB of the MYCIN is updated successfully. In order to test it, the doctor provides a new problem to it. The problem is to identify the presence of the bacteria by inputting the details of a patient, including the symptoms, current condition, and medical history.
- The ES will need a questionnaire to be filled by the patient to know the general information about the patient, such as gender, age, etc.
- Now the system has collected all the information, so it will find the solution for the problem by applying if-then rules using the inference engine and using the facts stored within the KB.
- In the end, it will provide a response to the patient by using the user interface.

Participants in the development of Expert System

There are three primary participants in the building of Expert System:

1. **Expert:** The success of an ES much depends on the knowledge provided by human experts. These experts are those persons who are specialized in that specific domain.
2. **Knowledge Engineer:** Knowledge engineer is the person who gathers the knowledge from the domain experts and then codifies that knowledge to the system according to the formalism.
3. **End-User:** This is a particular person or a group of people who may not be experts, and working on the expert system needs the solution or advice for his queries, which are complex.

Why Expert System?



Before using any technology, we must have an idea about why to use that technology and hence the same for the ES. Although we have human experts in every field, then what is the need to develop a computer-based system. So below are the points that are describing the need of the ES:

1. **No memory Limitations:** It can store as much data as required and can memorize it at the time of its application. But for human experts, there are some limitations to memorize all things at every time.
2. **High Efficiency:** If the knowledge base is updated with the correct knowledge, then it provides a highly efficient output, which may not be possible for a human.
3. **Expertise in a domain:** There are lots of human experts in each domain, and they all have different skills, different experiences, and different skills, so it is not easy to get a final output for the query. But if we put the knowledge gained from human experts into the expert system, then it provides an efficient output by mixing all the facts and knowledge

4. **Not affected by emotions:** These systems are not affected by human emotions such as fatigue, anger, depression, anxiety, etc.. Hence the performance remains constant.
5. **High security:** These systems provide high security to resolve any query.
6. **Considers all the facts:** To respond to any query, it checks and considers all the available facts and provides the result accordingly. But it is possible that a human expert may not consider some facts due to any reason.
7. **Regular updates improve the performance:** If there is an issue in the result provided by the expert systems, we can improve the performance of the system by updating the knowledge base.

Capabilities of the Expert System

Below are some capabilities of an Expert System:

- **Advising:** It is capable of advising the human being for the query of any domain from the particular ES.
- **Provide decision-making capabilities:** It provides the capability of decision making in any domain, such as for making any financial decision, decisions in medical science, etc.
- **Demonstrate a device:** It is capable of demonstrating any new products such as its features, specifications, how to use that product, etc.
- **Problem-solving:** It has problem-solving capabilities.
- **Explaining a problem:** It is also capable of providing a detailed description of an input problem.
- **Interpreting the input:** It is capable of interpreting the input given by the user.
- **Predicting results:** It can be used for the prediction of a result.
- **Diagnosis:** An ES designed for the medical field is capable of diagnosing a disease without using multiple components as it already contains various inbuilt medical tools.

Advantages of Expert System

- These systems are highly reproducible.
- They can be used for risky places where the human presence is not safe.

- Error possibilities are less if the KB contains correct knowledge.
- The performance of these systems remains steady as it is not affected by emotions, tension, or fatigue.
- They provide a very high speed to respond to a particular query.

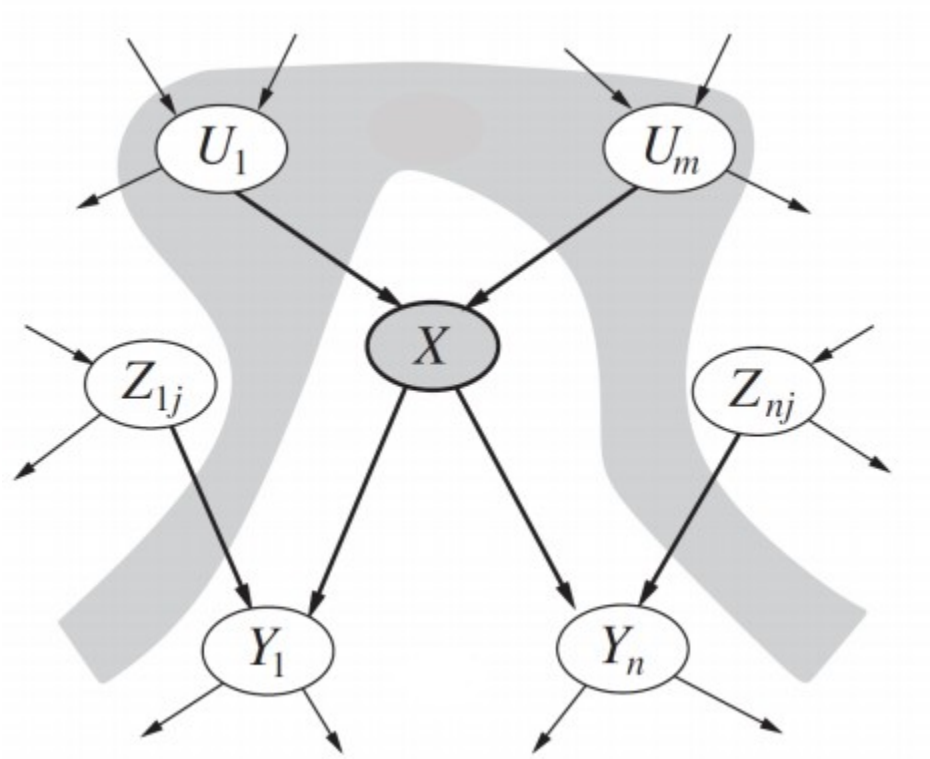
Limitations of Expert System

- The response of the expert system may get wrong if the knowledge base contains the wrong information.
- Like a human being, it cannot produce a creative output for different scenarios.
- Its maintenance and development costs are very high.
- Knowledge acquisition for designing is much difficult.
- For each domain, we require a specific ES, which is one of the big limitations.
- It cannot learn from itself and hence requires manual updates.

Applications of Expert System

- **In designing and manufacturing domain**
  It can be broadly used for designing and manufacturing physical devices such as camera lenses and automobiles.
- **In the knowledge domain**
  These systems are primarily used for publishing the relevant knowledge to the users. The two popular ES used for this domain is an advisor and a tax advisor.
- **In the finance domain**
  In the finance industries, it is used to detect any type of possible fraud, suspicious activity, and advise bankers that if they should provide loans for business or not.
- **In the diagnosis and troubleshooting of devices**
  In medical diagnosis, the ES system is used, and it was the first area where these systems were used.
- **Planning and Scheduling**
  The expert systems can also be used for planning and scheduling some particular tasks for achieving the goal of that task.

Bayesian Networks

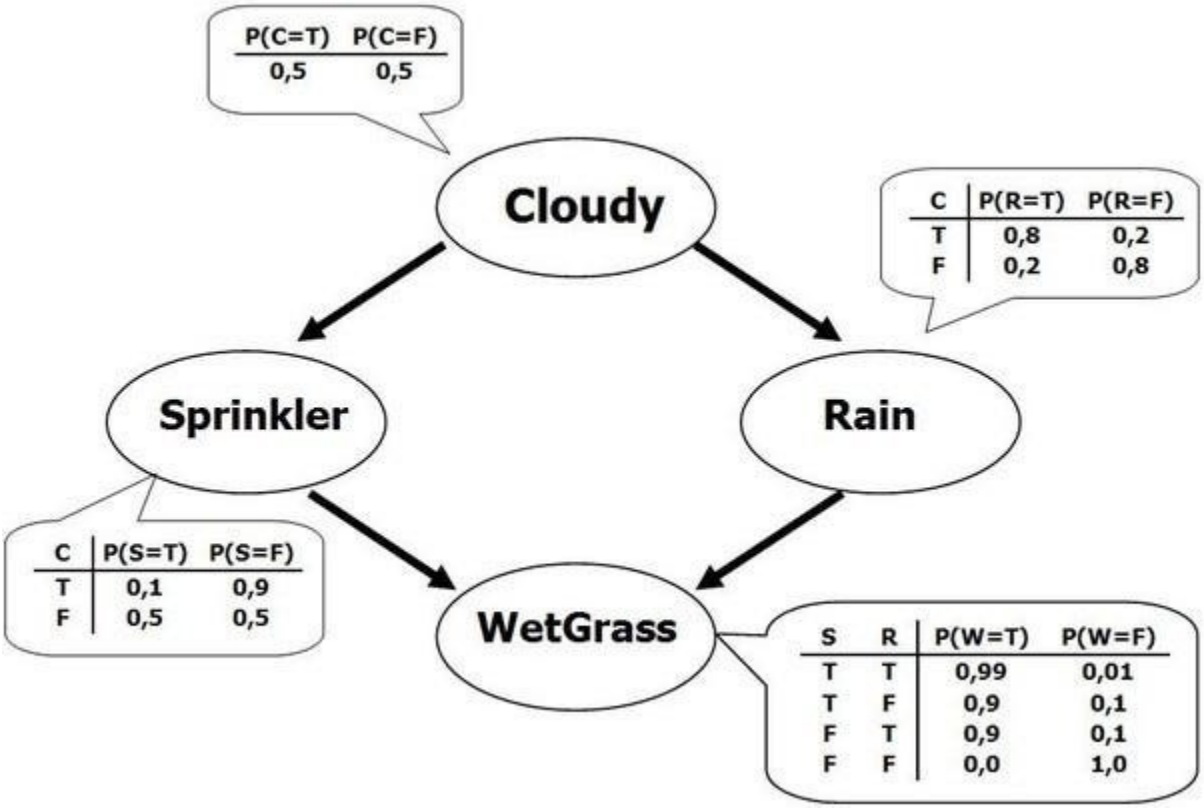


Introduction

Bayesian networks are a type of probabilistic graphical model which represents a set of variables and their conditional dependencies using a Directed Acyclic Graph (DAG). It uses Bayesian inference for probability computations. Bayesian networks aim to model conditional dependence, and therefore causation, by representing conditional dependence by edges in a directed graph. Through these relationships, one can efficiently conduct inference on the random variables in the graph through the use of factors.

The Bayesian Network

Using the relationships specified by our Bayesian network, we can obtain a compact, factorized representation of the joint probability distribution by taking advantage of conditional independence.



A Bayesian network is a **directed acyclic graph** in which each edge corresponds to a conditional dependency, and each node corresponds to a unique random variable. Formally, if an edge (A, B) exists in the graph connecting random variables A and B, it means that P(B | A) is a **factor** in the joint probability distribution, so we must know P(B | A) for all values of B and A in order to conduct inference. In the above example, since Rain has an edge going into WetGrass, it means that P(WetGrass | Rain) will be a factor, whose probability values are specified next to the WetGrass node in a conditional probability table.

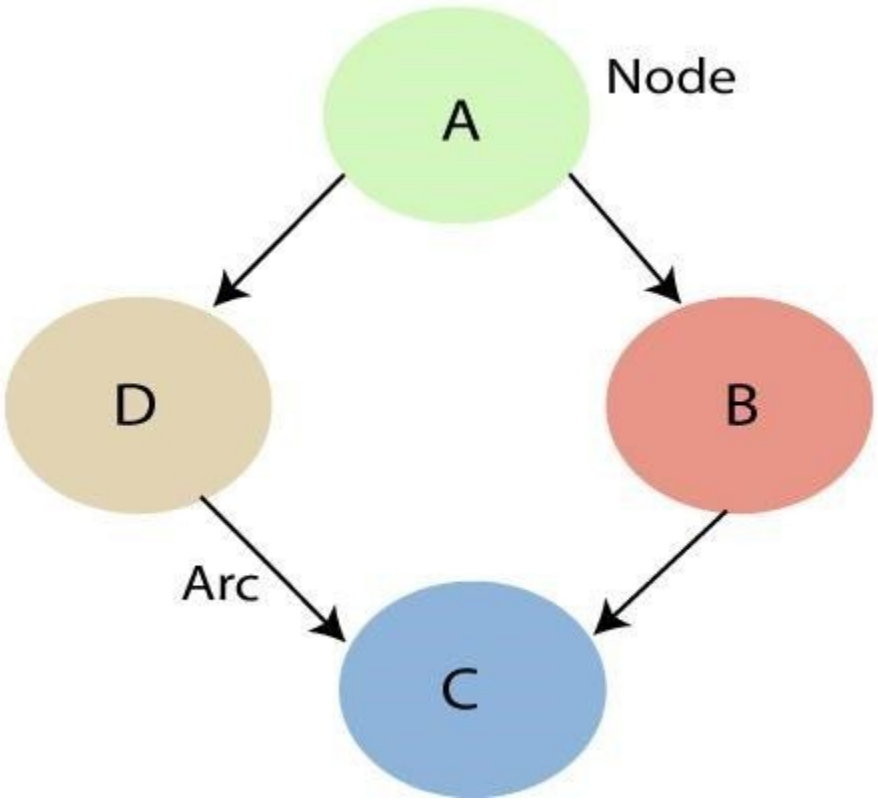
Bayesian networks satisfy the **local Markov property**, which states that a node is conditionally independent of its non-descendants given its parents. In the above example, this means that P(Sprinkler | Cloudy, Rain) = P(Sprinkler | Cloudy) since Sprinkler is conditionally independent of its non-descendant, Rain, given Cloudy. This property allows us to simplify the joint distribution, obtained in the previous section using the chain rule, to a smaller form. After simplification, the joint distribution for a Bayesian network is equal to the product of P(node | parents(node)) for all nodes, stated below:

$$P(X_1,\ldots,X_n)=\prod_{i=1}^{n}P(X_i \mid X_1,\ldots,X_{i-1})=\prod_{i=1}^{n}P(X_i \mid Parents(X_i))$$

In larger networks, this property allows us to greatly reduce the amount of required computation, since generally, most nodes will have few parents relative to the overall size of the network.

What are Bayesian Networks?

By definition, Bayesian Networks are a type of Probabilistic Graphical Model that uses the Bayesian inferences for probability computations. It represents a set of variables and its conditional probabilities with a Directed Acyclic Graph (DAG). They are primarily suited for considering an event that has occurred and predicting the likelihood that any one of the several possible known causes is the contributing factor.



As mentioned above, by making use of the relationships which are specified by the Bayesian Network, we can obtain the Joint Probability Distribution (JPF) with the conditional probabilities. Each node in the graph represents a random variable and the arc (or directed arrow) represents the relationship between the nodes. They can be either continuous or discrete in nature.

In the above diagram A, B, C and D are 4 random variables represented by nodes given in the network of the graph. To node B, A is its parent node and C is its child node. Node C is independent of Node A.

Before we get into the implementation of a Bayesian Network, there are a few probability basics that have to be understood.

Local Markov Property

The Bayesian Networks satisfy the property known as the Local Markov Property. It states that a node is conditionally independent of its non-descendants, given its parents. In the above example, P(D|A, B) is equal to P(D|A) because D is independent of its non-descendent, B. This property aids us in simplifying the Joint Distribution. The Local Markov Property leads us to the concept of a Markov Random Field which is a random field around a variable that is said to follow Markov properties.

Conditional Probability

In mathematics, the Conditional Probability of event A is the probability that event A will occur given that another event B has already occurred. In simple terms, p(A | B) is the probability of event A occurring, given that event, B occurs. However, there are two types of event possibilities between A and B. They may be either dependent events or independent events. Depending upon their type, there are two different ways to calculate the conditional probability.

- Given A and B are dependent events, the conditional probability is calculated as P (A| B) = P (A and B) / P (B)
- If A and B are independent events, then the expression for conditional probability is given by, P(A| B) = P (A)

Joint Probability Distribution

Before we get into an example of Bayesian Networks, let us understand the concept of Joint Probability Distribution. Consider 3 variables a1, a2 and a3. By definition, the probabilities of all different possible combinations of a1, a2, and a3 are called its Joint Probability Distribution.

If P[a1,a2, a3,….., an] is the JPD of the following variables from a1 to an, then there are several ways of calculating the Joint Probability Distribution as a combination of various terms such as,

P[a1,a2, a3,….., an] = P[a1 | a2, a3,….., an] * P[a2, a3,….., an]

= P[a1 | a2, a3,….., an] * P[a2 | a3,….., an]….P[an-1|an] * P[an]

Generalizing the above equation, we can write the Joint Probability Distribution as,

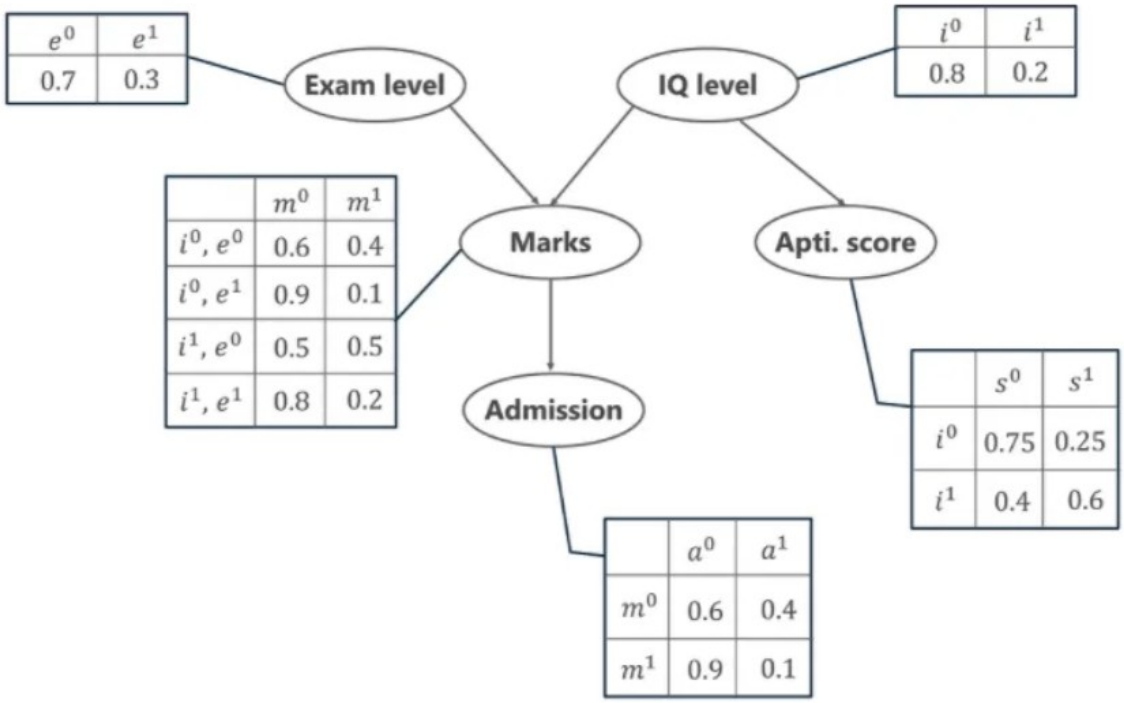
P(Xi|Xi-1,……., Xn) = P(Xi |Parents(Xi))

Bayesian Network Example

Q1. Let us now understand the mechanism of Bayesian Networks and their advantages with the help of a simple example. In this example, let us imagine that we are given the task of modeling a student's marks (***m***) for an exam he has just given. From the given Bayesian Network Graph below, we see that the marks depend upon two other variables. They are,

- Exam Level (***e***)– This discrete variable denotes the difficulty of the exam and has two values (0 for easy and 1 for difficult)
- IQ Level (***i***) – This represents the Intelligence Quotient level of the student and is also discrete in nature having two values (0 for low and 1 for high)

Additionally, the IQ level of the student also leads us to another variable, which is the Aptitude Score of the student (***s***). Now, with marks the student has scored, he can secure admission to a particular university. The probability distribution for getting admitted (***a***) to a university is also given below.



In the above graph, we see several tables representing the probability distribution values of the given 5 variables. These tables are called the Conditional Probabilities Table or CPT. There are a few properties of the CPT given below –

- The sum of the CPT values in each row must be equal to 1 because all the possible cases for a particular variable are exhaustive (representing all possibilities).
- If a variable that is Boolean in nature has k Boolean parents, then in the CPT it has 2K probability values.

Coming back to our problem, let us first list all the possible events that are occurring in the above-given table.

1. Exam Level (e)
2. IQ Level (i)
3. Aptitude Score (s)
4. Marks (m)

5. Admission (a)

These five variables are represented in the form of a Directed Acyclic Graph (DAG) in a Bayesian Network format with their Conditional Probability tables. Now, to calculate the Joint Probability Distribution of the 5 variables the formula is given by,

P[a, m, i, e, s]= P(a | m) . P(m | i, e) . P(i) . P(e) . P(s | i)

From the above formula,

- P(a | m) denotes the conditional probability of the student getting admission based on the marks he has scored in the examination.
- P(m | i, e) represents the marks that the student will score given his IQ level and difficulty of the Exam Level.
- P(i) and P(e) represent the probability of the IQ Level and the Exam Level.
- P(s | i) is the conditional probability of the student's Aptitude Score, given his IQ Level.

With the following probabilities calculated, we can find the Joint Probability Distribution of the entire Bayesian Network.

Calculation of Joint Probability Distribution

Let us now calculate the JPD for two cases.

Case 1: Calculate the probability that in spite of the exam level being difficult, the student having a low IQ level and a low Aptitude Score, manages to pass the exam and secure admission to the university.

From the above word problem statement, the Joint Probability Distribution can be written as below,

P[a=1, m=1, i=0, e=1, s=0]

From the above Conditional Probability tables, the values for the given conditions are fed to the formula and is calculated as below.

P[a=1, m=1, i=0, e=0, s=0] = P(a=1 | m=1) . P(m=1 | i=0, e=1) . P(i=0) . P(e=1) . P(s=0 | i=0)

= 0.1 * 0.1 * 0.8 * 0.3 * 0.75

= **0.0018**

Case 2: In another case, calculate the probability that the student has a High IQ level and Aptitude Score, the exam being easy yet fails to pass and does not secure admission to the university.

The formula for the JPD is given by

P[a=0, m=0, i=1, e=0, s=1]

Thus,

P[a=0, m=0, i=1, e=0, s=1]= P(a=0 | m=0) . P(m=0 | i=1, e=0) . P(i=1) . P(e=0) . P(s=1 | i=1)

= 0.6 * 0.5 * 0.2 * 0.7 * 0.6

= **0.0252**

Hence, in this way, we can make use of Bayesian Networks and Probability tables to calculate the probability for various possible events that occur.

Q2. **Example:** Harry installed a new burglar alarm at his home to detect burglary. The alarm reliably responds at detecting a burglary but also responds for minor earthquakes. Harry has two neighbors David and Sophia, who have taken a responsibility to inform Harry at work when they hear the alarm. David always calls Harry when he hears the alarm, but sometimes he got confused with the phone ringing and calls at that time too. On the other hand, Sophia likes to listen to high music, so sometimes she misses to hear the alarm. Here we would like to compute the probability of Burglary Alarm.

Problem:

Calculate the probability that alarm has sounded, but there is neither a burglary, nor an earthquake occurred, and David and Sophia both called the Harry.

Solution:

- The Bayesian network for the above problem is given below. The network structure is showing that burglary and earthquake is the parent node of the alarm and directly affecting the probability of alarm's going off, but David and Sophia's calls depend on alarm probability.
- The network is representing that our assumptions do not directly perceive the burglary and also do not notice the minor earthquake, and they also not confer before calling.
- The conditional distributions for each node are given as conditional probabilities table or CPT.
- Each row in the CPT must be sum to 1 because all the entries in the table represent an exhaustive set of cases for the variable.
- In CPT, a boolean variable with k boolean parents contains 2^K probabilities. Hence, if there are two parents, then CPT will contain 4 probability values

List of all events occurring in this network:

- **Burglary (B)**
- **Earthquake(E)**
- **Alarm(A)**
- **David Calls(D)**
- **Sophia calls(S)**

We can write the events of problem statement in the form of probability: **P[D, S, A, B, E]**, can rewrite the above probability statement using joint probability distribution:

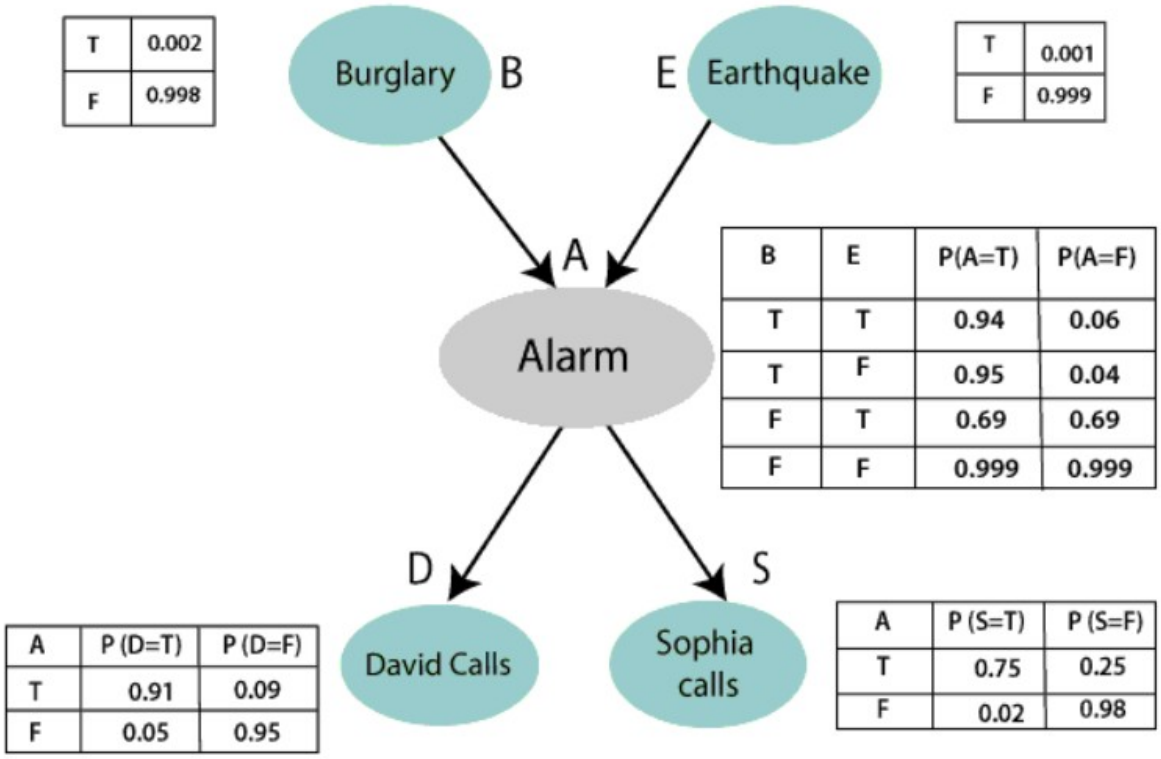
P[D, S, A, B, E]= P[D | S, A, B, E]. P[S, A, B, E]

=P[D | S, A, B, E]. P[S | A, B, E]. P[A, B, E]

= P [D| A]. P [S| A, B, E]. P[A, B, E]

= P[D | A]. P[S | A]. P[A| B, E]. P[B, E]

= P[D | A]. P[S | A]. P[A| B, E]. P[B |E]. P[E]



T	0.002
F	0.998

T	0.001
F	0.999

B	E	P(A=T)	P(A=F)
T	T	0.94	0.06
T	F	0.95	0.04
F	T	0.69	0.69
F	F	0.999	0.999

A	P (D=T)	P (D=F)
T	0.91	0.09
F	0.05	0.95

A	P (S=T)	P (S=F)
T	0.75	0.25
F	0.02	0.98

Let's take the observed probability for the Burglary and earthquake component:

P(B= True) = 0.002, which is the probability of burglary.

P(B= False)= 0.998, which is the probability of no burglary.

P(E= True)= 0.001, which is the probability of a minor earthquake

P(E= False)= 0.999, Which is the probability that an earthquake not occurred.

We can provide the conditional probabilities as per the below tables:

Conditional probability table for Alarm A:

The Conditional probability of Alarm A depends on Burglar and earthquake:

B	E	P(A= True)	P(A= False)
True	True	0.94	0.06
True	False	0.95	0.04
False	True	0.31	0.69
False	False	0.001	0.999

Conditional probability table for David Calls:

The Conditional probability of David that he will call depends on the probability of Alarm.

A	P(D= True)	P(D= False)
True	0.91	0.09
False	0.05	0.95

Conditional probability table for Sophia Calls:

The Conditional probability of Sophia that she calls is depending on its Parent Node "Alarm."

A	P(S= True)	P(S= False)
True	0.75	0.25
False	0.02	0.98

From the formula of joint distribution, we can write the problem statement in the form of probability distribution:

P(S, D, A, ¬B, ¬E) = P (S|A) *P (D|A)*P (A|¬B ^ ¬E) *P (¬B) *P (¬E).

= 0.75* 0.91* 0.001* 0.998*0.999

= **0.00068045.**

Hence, a Bayesian network can answer any query about the domain by using Joint distribution.

The semantics of Bayesian Network:

There are two ways to understand the semantics of the Bayesian network, which is given below:

1. To understand the network as the representation of the Joint probability distribution.

It is helpful to understand how to construct the network.

2. To understand the network as an encoding of a collection of conditional independence statements.

It is helpful in designing inference procedure.

K-Means Clustering Algorithm

K-means clustering is a popular method for grouping data by assigning observations to clusters based on proximity to the cluster's center. This article explores k-means clustering, its importance, applications, and workings, providing a clear understanding of its role in data analysis. In this article, you will explore k-means clustering, an unsupervised learning technique that groups data points into clusters based on similarity. A k means clustering example illustrates how this method assigns data points to the nearest centroid, refining the clusters iteratively. Understanding what is k-means clustering will enhance your grasp of **data analysis** and pattern recognition.

What is K-Means Clustering?

K-means clustering is a popular unsupervised **machine learning algorithm** used for partitioning a dataset into a pre-defined number of clusters. The goal is to group similar data points together and discover underlying patterns or structures within the data.

- Recall the first property of clusters – it states that the points within a cluster should be similar to each other. So, our aim here is to minimize the distance between the points within a cluster.
- There is an algorithm that tries to minimize the distance of the points in a cluster with their centroid – the k-means clustering technique.
- K-means is a centroid-based algorithm or a distance-based algorithm, where we calculate the distances to assign a point to a cluster. In K-Means, each cluster is associated with a centroid.

The main objective of the K-Means algorithm is to minimize the sum of distances between the points and their respective cluster centroid.

Optimization plays a crucial role in the k-means clustering algorithm. The goal of the optimization process is to find the best set of centroids that minimizes the sum of squared distances between each data point and its closest centroid.

How K-Means Clustering Works?

Here's how it works:

1. **Initialization**: Start by randomly selecting K points from the dataset. These points will act as the initial cluster centroids.
2. **Assignment**: For each data point in the dataset, calculate the distance between that point and each of the K centroids. Assign the data point to the cluster whose centroid is closest to it. This step effectively forms K clusters.
3. **Update centroids**: Once all data points have been assigned to clusters, recalculate the centroids of the clusters by taking the mean of all data points assigned to each cluster.
4. **Repeat**: Repeat steps 2 and 3 until convergence. Convergence occurs when the centroids no longer change significantly or when a specified number of iterations is reached.
5. **Final Result**: Once convergence is achieved, the algorithm outputs the final cluster centroids and the assignment of each data point to a cluster.

Objective of k means Clustering

The main objective of k-means clustering is to partition your data into a specific number (k) of groups, where data points within each group are similar and dissimilar to points in other groups. It achieves this by minimizing the distance between data points and their assigned cluster's center, called the centroid.

Here's an objective:

- **Grouping similar data points:** K-means aims to identify patterns in your data by grouping data points that share similar characteristics together. This allows you to discover underlying structures within the data.
- **Minimizing within-cluster distance:** The algorithm strives to make sure data points within a cluster are as close as possible to each other, as measured by a distance metric (usually Euclidean distance). This ensures tight-knit clusters with high cohesiveness.
- **Maximizing between-cluster distance:** Conversely, k-means also tries to maximize the separation between clusters. Ideally, data points from

different clusters should be far apart, making the clusters distinct from each other.

What is Clustering?

Cluster analysis is a technique in **data mining** and **machine learning** that groups similar objects into clusters. K-means clustering, a popular method, aims to divide a set of objects into K clusters, minimizing the sum of squared distances between the objects and their respective cluster centers.

Hierarchical clustering and k-means clustering are two popular techniques in the field of unsupervised learning used for clustering data points into distinct groups. While k-means clustering divides data into a predefined number of clusters, hierarchical clustering creates a hierarchical tree-like structure to represent the relationships between the clusters.

Example of Clustering

Let's try understanding this with a simple example. A bank wants to give credit card offers to its customers. Currently, they look at the details of each customer and, based on this information, decide which offer should be given to which customer.

Now, the bank can potentially have millions of customers. Does it make sense to look at the details of each customer separately and then make a decision? Certainly not! It is a manual process and will take a huge amount of time.

So what can the bank do? One option is to segment its customers into different groups. For instance, the bank can group the customers based on their income:

Can you see where I'm going with this? The bank can now make three different strategies or offers, one for each group. Here, instead of creating different strategies for individual customers, they only have to make 3 strategies. This will reduce the effort as well as the time.

The groups I have shown above are known as clusters, and the process of creating these groups is known as clustering. Formally, we can say that:

- Clustering is the process of dividing the entire data into groups (also known as clusters) based on the patterns in the data.
- Can you guess which type of learning problem clustering is? Is it a **supervised or unsupervised learning** problem?

Think about it for a moment and use the example we just saw. Got it? Clustering is an unsupervised learning problem!

How is Clustering an Unsupervised Learning Problem?

Let's say you are working on a project where you need to predict the sales of a big mart:

Outlet_Size	Outlet_Location_Type	Outlet_Type	Item_Outlet_Sales
Medium	Tier 1	Supermarket Type1	3735.1380
Medium	Tier 3	Supermarket Type2	443.4228
Medium	Tier 1	Supermarket Type1	2097.2700
NaN	Tier 3	Grocery Store	732.3800
High	Tier 3	Supermarket Type1	994.7052

Or, a project where your task is to predict whether a loan will be approved or not:

Loan_ID	Gender	Married	ApplicantIncome	LoanAmount	Loan_Status
LP001002	Male	No	5849	130.0	Y
LP001003	Male	Yes	4583	128.0	N
LP001005	Male	Yes	3000	66.0	Y
LP001006	Male	Yes	2583	120.0	Y
LP001008	Male	No	6000	141.0	Y

We have a fixed target to predict in both of these situations. In the sales prediction problem, we have to predict the *Item_Outlet_Sales* based on *outlet_size, outlet_location_type*, etc., and in the loan approval problem, we have to predict the *Loan_Status* depending on the Gender, marital status, the income of the customers, etc.

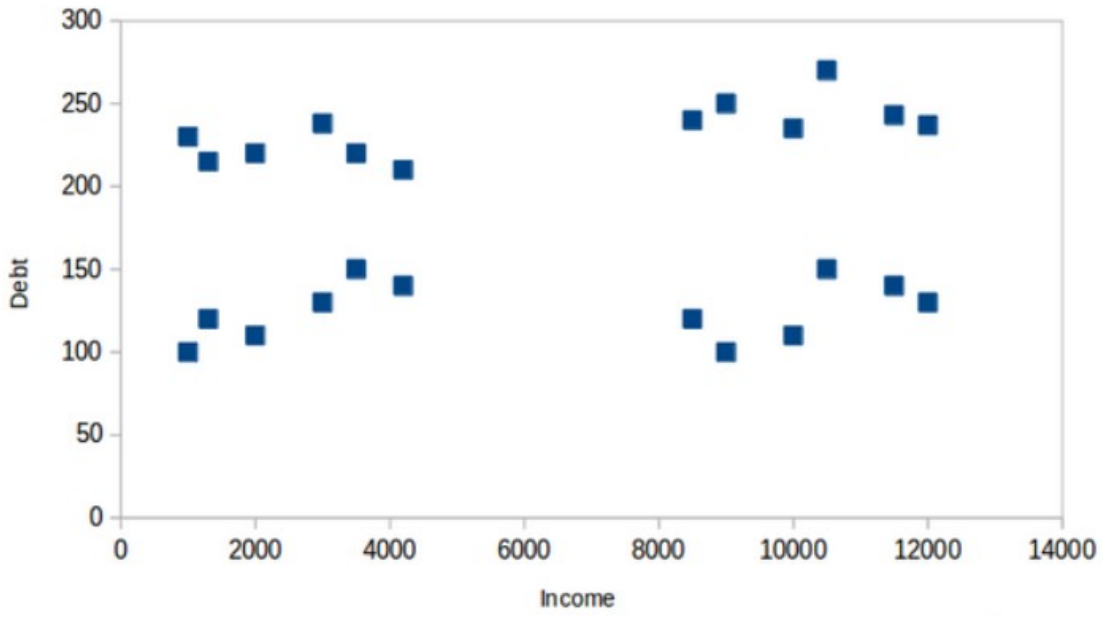
- So, when we have a target variable to predict based on a given set of predictors or independent variables, such problems are called supervised learning problems.
- Now, there might be situations where we do *not* have any target variable to predict.
- Such problems, without any fixed target variable, are known as unsupervised learning problems. In these problems, we only have the independent variables and no target/dependent variable.

In clustering, we do not have a target to predict. We look at the data, try to club similar observations, and form different groups. Hence it is an unsupervised learning problem.

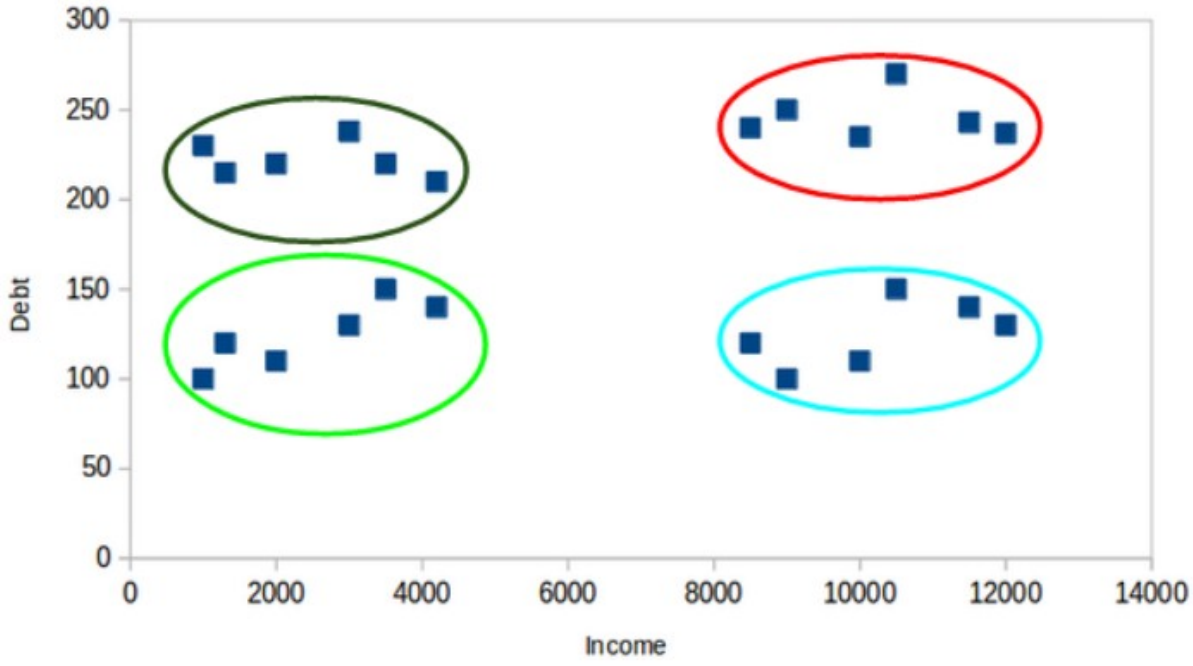
We now know what clusters are and the concept of clustering. Next, let's look at the properties of these clusters, which we must consider while forming the clusters.

Properties of K means Clustering

How about another example of k-means clustering algorithm? We'll take the same bank as before, which wants to segment its customers. For simplicity purposes, let's say the bank only wants to use the income and debt to make the **segmentation**. They collected the customer data and used a scatter plot to visualize it:



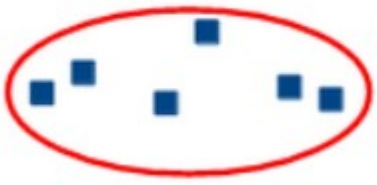
On the X-axis, we have the income of the customer, and the y-axis represents the amount of debt. Here, we can clearly visualize that these customers can be segmented into 4 different clusters, as shown below:



This is how clustering helps to create segments (clusters) from the data. The bank can further use these clusters to make strategies and offer discounts to its customers. So let's look at the properties of these clusters.

First Property of K-Means Clustering Algorithm

All the data points in a cluster should be similar to each other. Let me illustrate it using the above example:

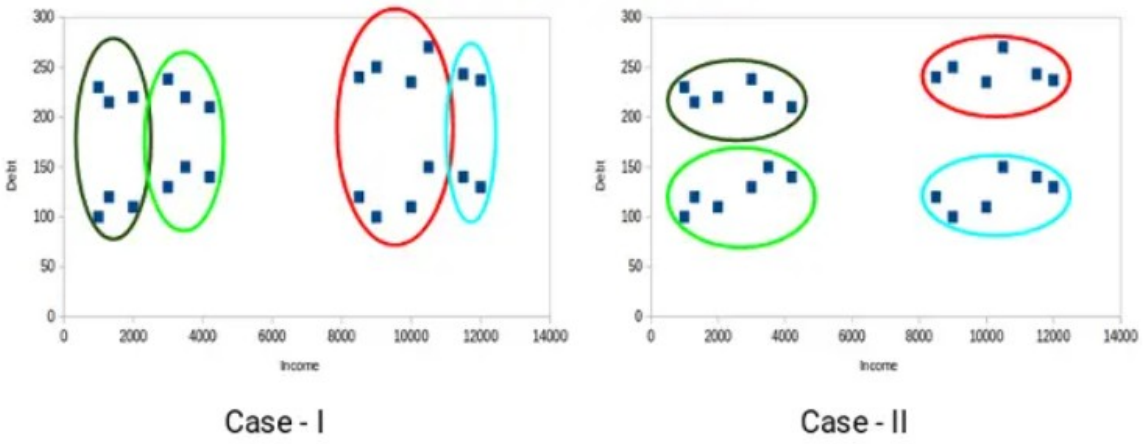


If the customers in a particular cluster are not similar to each other, then their requirements might vary, right? If the bank gives them the same offer, they might not like it, and their interest in the bank might reduce. Not ideal.

Having similar data points within the same cluster helps the bank to use targeted marketing. You can think of similar examples from your everyday life and consider how clustering will (or already does) impact the business strategy.

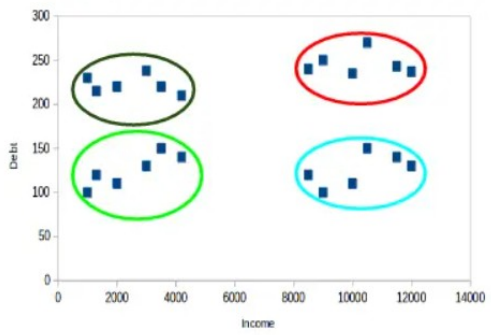
Second Property of K-Means Clustering Algorithm

The data points from different clusters should be as different as possible. This will intuitively make sense if you've grasped the above property. Let's again take the same example to understand this property:



Which of these cases do you think will give us the better clusters? If you look at case I:

Customers in the red and blue clusters are quite similar to each other. The top four points in the red cluster share similar properties to those of the blue cluster's top two customers. They have high incomes and high debt values. Here, we have clustered them differently. Whereas, if you look at case II:



Case - II

Points in the red cluster completely differ from the customers in the blue cluster. All the customers in the red cluster have high income and high debt, while the customers in the blue cluster have high income and low debt value. Clearly, we have a better clustering of customers in this case.

Hence, data points from different clusters should be as different from each other as possible to have more meaningful clusters. The k-means algorithm uses an iterative approach to find the optimal cluster assignments by minimizing the sum of squared distances between data points and their assigned cluster centroid

MACHINE LEARNING Notes - 201CS6T01

Unit – I

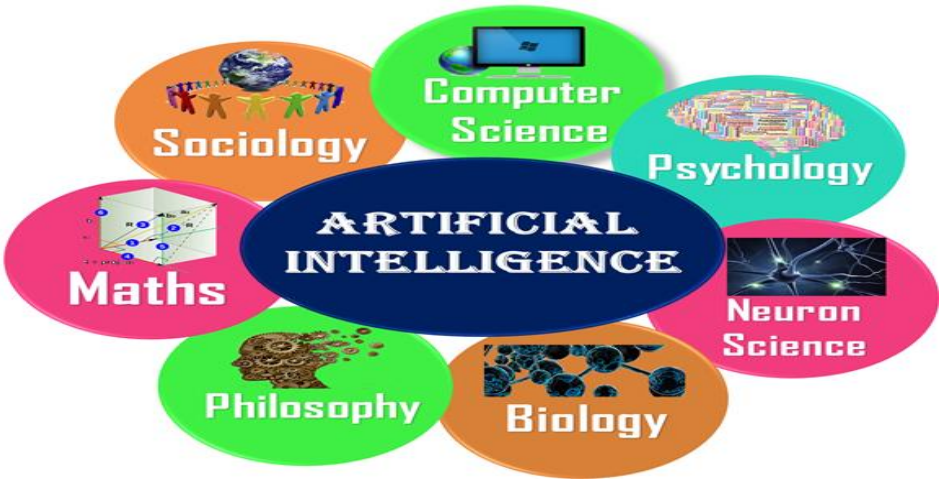
Introduction- Artificial Intelligence, Machine Learning, Deep learning, Types of Machine Learning Systems, Main Challenges of Machine Learning. Statistical Learning: Introduction, Supervised and Unsupervised Learning, Training and Test Loss, Trade-offs in Statistical Learning, Estimating Risk Statistics, Sampling distribution of an estimator, Empirical Risk Minimization.

TOPIC-1: Introduction- Artificial Intelligence, Machine Learning, Deep learning:

- **Artificial Intelligence (AI):** In today's world, technology is growing very fast, and we are getting in touch with different new technologies day by day.
- Here, one of the booming technologies of computer science is Artificial Intelligence which is ready to create a new revolution in the world by making intelligent machines.
- Artificial Intelligence is composed of two words Artificial and Intelligence, where Artificial defines "man-made," and intelligence defines "thinking power", hence AI means "a man-made thinking power."
- So, we can define AI as: "It is a branch of computer science by which we can create intelligent machines which can behave like a human, think like humans, and able to make decisions."
- Artificial Intelligence exists when a machine can have human based skills such as learning, reasoning, and solving problems.

Why Artificial Intelligence?

- With the help of AI, you can create such software or devices which can solve real-world problems very easily and with accuracy such as health issues, marketing, traffic issues, etc.
- With the help of AI, you can create your personal virtual Assistant, such as Cortana, Google Assistant, Siri, etc.
- With the help of AI, you can build such Robots which can work in an environment where survival of humans can be at risk.
- AI opens a path for other new technologies, new devices, and new Opportunities.



Machine Learning:

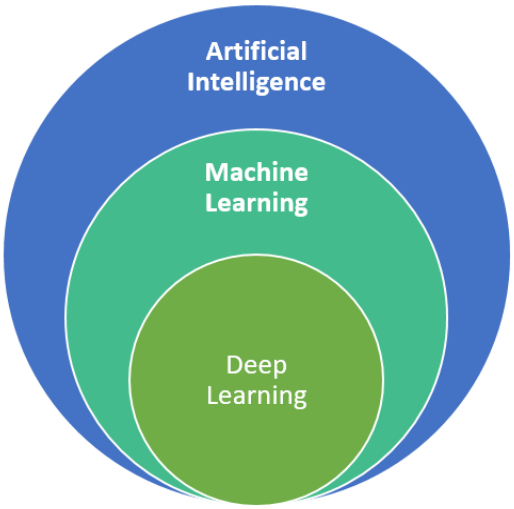
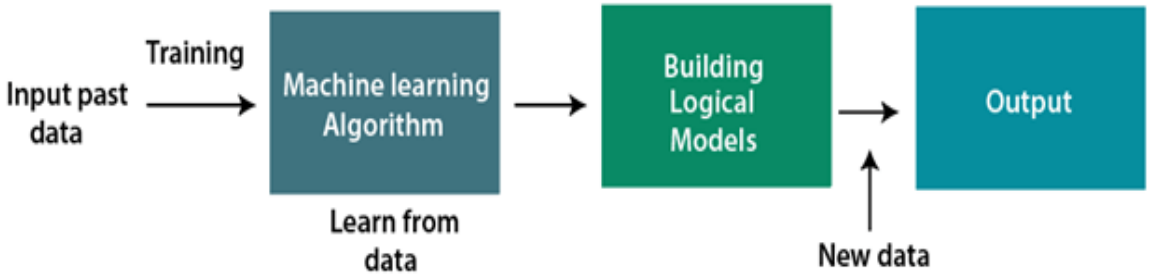


Figure 1: artificial intelligence, machine leaning and deep learning Source: Nadia BERCHANE (M2 IESCI, 2018)

- Machine learning is a growing technology which enables computers to learn automatically from past data.
- Machine learning uses various algorithms for building mathematical models and making predictions using historical data or information.
- Currently, it is being used for various tasks such as image recognition, speech recognition, email filtering, Facebook auto-tagging, recommender system, and many more.

Arthur Samuel

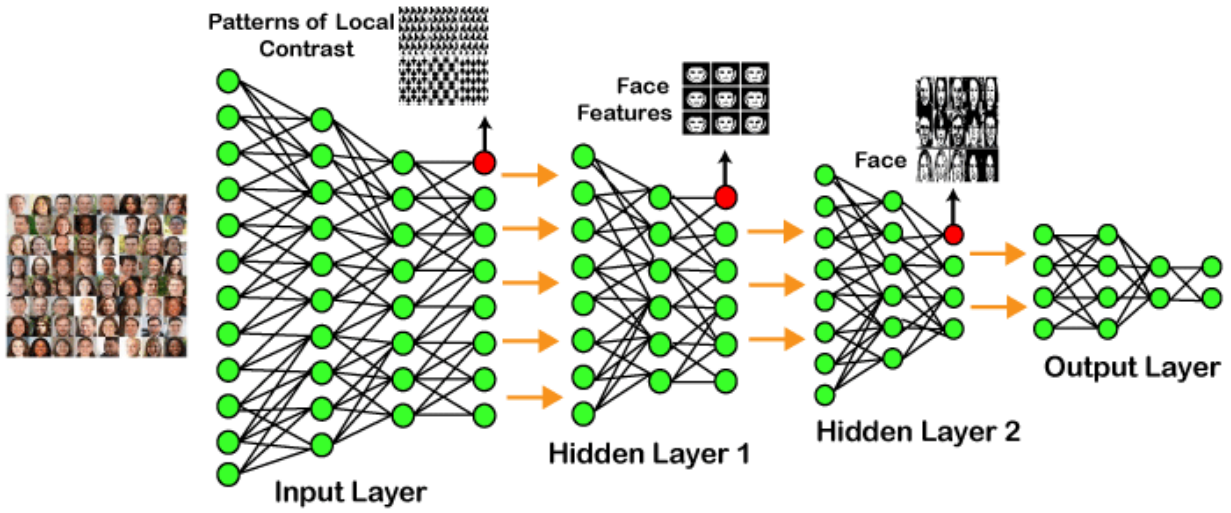
- The term machine learning was first introduced by Arthur Samuel in 1959. We can define it in a summarized way as:
- **Machine learning enables a machine to automatically learn from data, improve performance from experiences, and predict things without being explicitly programmed.**



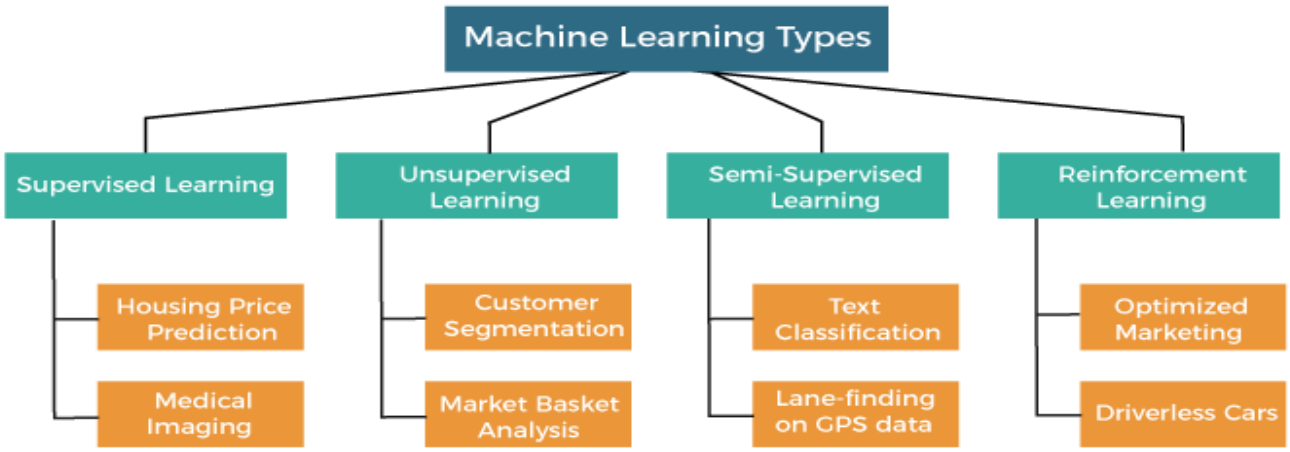
Deep Learning:

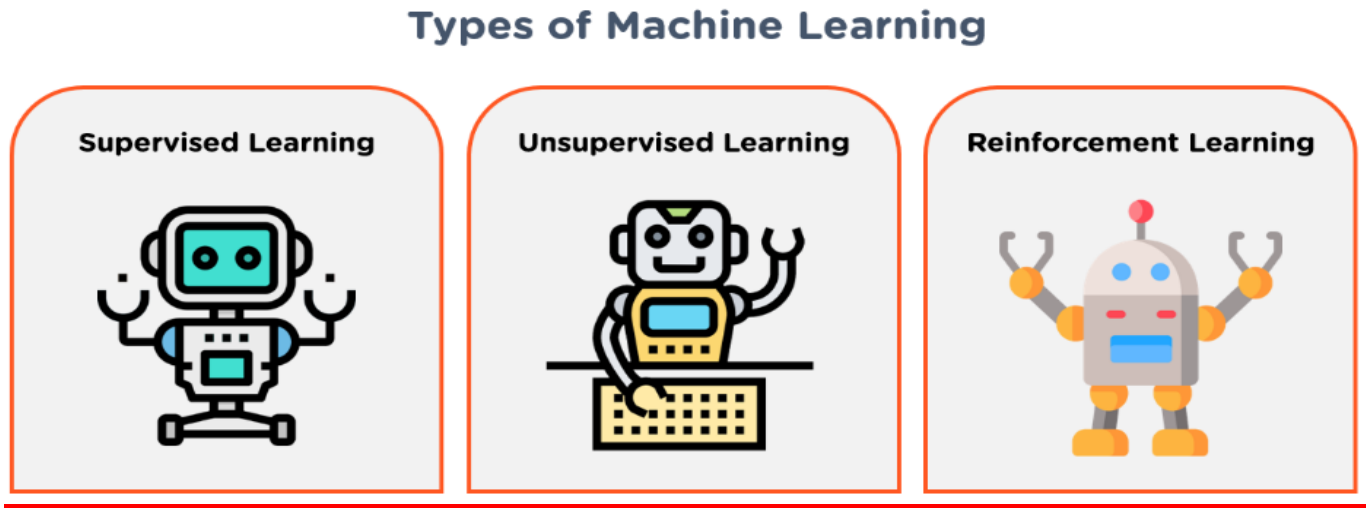
- Deep learning is based on the branch of machine learning, which is a subset of artificial intelligence.

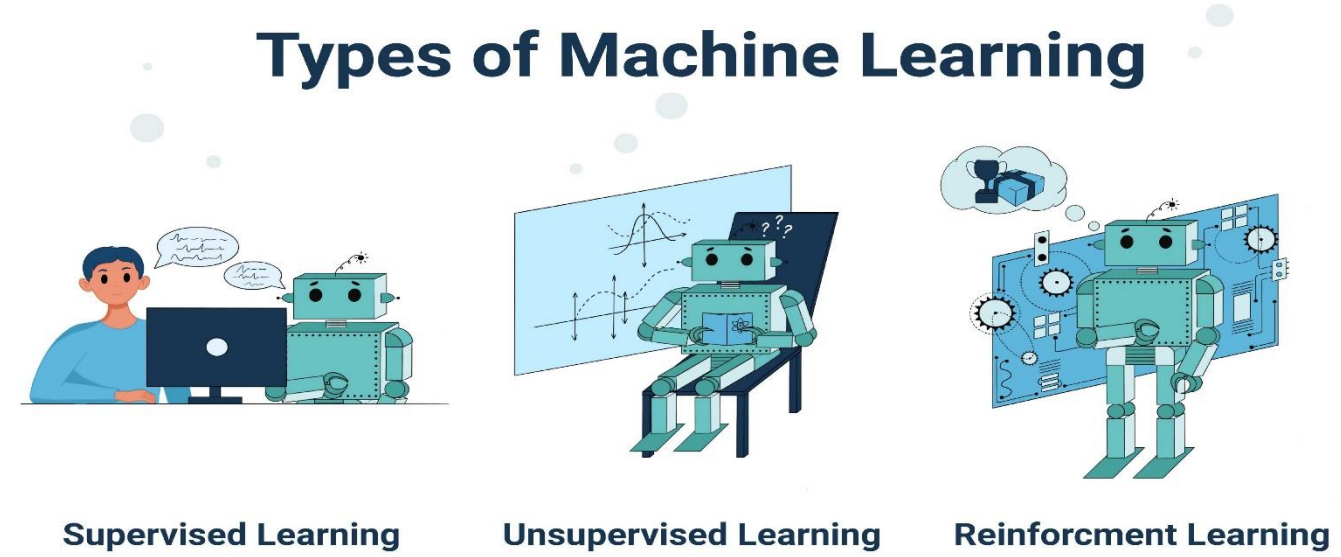
- Since neural networks imitate the human brain and so deep learning will do. In deep learning, nothing is programmed explicitly.
- Basically, it is a machine learning class that makes use of numerous nonlinear processing units so as to perform feature extraction as well as transformation.
- IDEA: Deep learning is implemented with the help of Neural Networks, and the idea behind the motivation of Neural Network is the biological neurons, which is nothing but a brain cell.
- Deep learning is a collection of statistical techniques of machine learning for learning feature hierarchies that are actually based on artificial neural networks.
- **Example of Deep Learning:**



TOPIC-2: Types of Machine Learning Systems







There are so many different types of Machine Learning systems that it is useful to classify them in broad categories, based on the following criteria:

1. Whether or not they are trained with human supervision (supervised, unsupervised, semi supervised, and Reinforcement Learning)

2. Whether or not they can learn incrementally on the fly (online versus batch learning)

3.Whether they work by simply comparing new data points to known data points, or instead by detecting patterns in the training data and building a predictive model, much like scientists do (instance-based versus model-based learning).

1. Supervised Machine Learning: As its name suggests, supervised machine learning is based on supervision.

- It means in the supervised learning technique, we train the machines using the "labelled" dataset, and based on the training, the machine predicts the output.
- The main goal of the supervised learning technique is to map the input variable(x) with the output variable(y). Some real-world applications of supervised learning are Risk Assessment, Fraud Detection, Spam filtering, etc.

Categories of Supervised Machine Learning:

- Supervised machine learning can be classified into two types of problems, which are given below:
- **Classification**
- **Regression**

Classification: Classification algorithms are used to solve the classification problems in which the output variable is categorical, such as "Yes" or No, Male or Female, Red or Blue, etc.

- The classification algorithms predict the categories present in the dataset.

- Some real-world examples of classification algorithms are Spam Detection, Email filtering, etc.

Some popular classification algorithms are given below:

- Random Forest Algorithm
- Decision Tree Algorithm
- Logistic Regression Algorithm
- Support Vector Machine Algorithm

Regression:

- Regression algorithms are used to solve regression problems in which there is a linear relationship between input and output variables.
- These are used to predict continuous output variables, such as market trends, weather prediction, etc.

Some popular Regression algorithms are given below:

- Simple Linear Regression Algorithm
- Multivariate Regression Algorithm
- Decision Tree Algorithm
- Lasso Regression

Advantages and Disadvantages of Supervised Learning:

Advantages:

- Since supervised learning work with the labelled dataset so we can have an exact idea about the classes of objects.
- These algorithms are helpful in predicting the output on the basis of prior experience.

Disadvantages:

- These algorithms are not able to solve complex tasks.
- It may predict the wrong output if the test data is different from the training data.
- It requires lots of computational time to train the algorithm.

2. Unsupervised Machine Learning:

- Unsupervised learning is different from the supervised learning technique; as its name suggests, there is no need for supervision.
- It means, in unsupervised machine learning, the machine is trained using the unlabeled dataset, and the machine predicts the output w
- **The main aim of the unsupervised learning algorithm is to group or categories the unsorted dataset according to the similarities, patterns, and differences.**
- Machines are instructed to find the hidden patterns from the input dataset.

Categories of Unsupervised Machine Learning:

Unsupervised Learning can be further classified into two types, which are given below:

- **Clustering**
- **Association**

1) Clustering:

- The clustering technique is used when we want to find the inherent groups from the data.
- It is a way to group the objects into a cluster such that the objects with the most similarities remain in one group and have fewer or no similarities with the objects of other groups.
- An example of the clustering algorithm is grouping the customers by their purchasing behavior.

Some of the popular clustering algorithms are given below:

- K-Means Clustering algorithm
- Mean-shift algorithm
- DBSCAN Algorithm
- Principal Component Analysis
- Independent Component Analysis

2) Association:

- Association rule learning is an unsupervised learning technique, which finds interesting relations among variables within a large dataset.
- The main aim of this learning algorithm is to find the dependency of one data item on another data item and map those variables accordingly so that it can generate maximum profit.
- Some popular algorithms of Association rule learning are **Apriori Algorithm, Eclat, FP-growth algorithm.**

Advantages and Disadvantages of Unsupervised Learning Algorithm:

Advantages:

- These algorithms can be used for complicated tasks compared to the supervised ones because these algorithms work on the unlabeled dataset.
- Unsupervised algorithms are preferable for various tasks as getting the unlabeled dataset is easier as compared to the labelled dataset.

Disadvantages:

- The output of an unsupervised algorithm can be less accurate as the dataset is not labelled, and algorithms are not trained with the exact output in prior.
- Working with Unsupervised learning is more difficult as it works with the unlabeled dataset that does not map with the output.

3. Semi-Supervised Learning:

- **Semi-Supervised learning is a type of Machine Learning algorithm that lies between Supervised and Unsupervised machine learning**.
- It represents the intermediate ground between Supervised (With Labelled training data) and Unsupervised learning (with no labelled training data) algorithms and uses the combination of labelled and unlabeled datasets during the training period.

To overcome the drawbacks of supervised learning and unsupervised learning algorithms, the concept of Semi-supervised learning is introduced.

- We can imagine these algorithms with an example. Supervised learning is where a student is under the supervision of an instructor at home and college.
- Further, if that student is self- analyzing the same concept without any help from the instructor, it comes under unsupervised learning.
- Under semi-supervised learning, the student has to revise himself after analyzing the same concept under the guidance of an instructor at college.

Advantages:

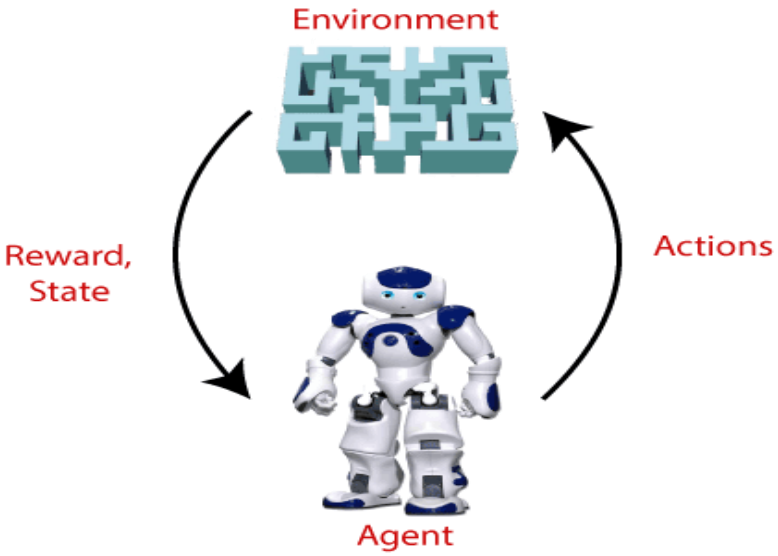
- It is simple and easy to understand the algorithm.
- It is highly efficient.
- It is used to solve drawbacks of Supervised and Unsupervised Learning algorithms.

Disadvantages:

- Iterations results may not be stable.
- We cannot apply these algorithms to network-level data.
- Accuracy is low.

4. Reinforcement Learning:

- **Reinforcement learning works on a feedback-based process, in which an AI agent (A software component) automatically explore its surrounding by hitting & trail, taking action, learning from experiences, and improving its performance.**
- Agent gets rewarded for each good action and get punished for each bad action; hence the goal of reinforcement learning agent is to maximize the rewards.
- In reinforcement learning, there is no labelled data like supervised learning, and agents learn from their experiences only.



- The reinforcement learning process is similar to a human being; for example, a child learns various things by experiences in his day-to-day life.
- An example of reinforcement learning is to play a game, where the Game is the environment, moves of an agent at each step define states, and the goal of the agent is to get a high score.
- Agent receives feedback in terms of punishment and rewards.
- Due to its way of working, reinforcement learning is employed in different fields such as **Game theory, Operation Research, Information theory, multi-agent systems.**

Categories of Reinforcement Learning:

- Reinforcement learning is categorized mainly into two types of methods/algorithms:
- **Positive Reinforcement Learning:** Positive reinforcement learning specifies increasing the tendency that the required behavior would occur again by adding something. It enhances the strength of the behavior of the agent and positively impacts it.
- **Negative Reinforcement Learning:** Negative reinforcement learning works exactly opposite to the positive RL. It increases the tendency that the specific behavior would occur again by avoiding the negative condition.

Real-world Use cases of Reinforcement Learning

- **Video Games**
- **Robotics**
- **Text Mining**

TOPIC-3: Main Challenges of Machine Learning:

1) Lack Of Quality Data

One of the main issues in Machine Learning is the absence of good data. While upgrading, algorithms tend to make developers exhaust most of their time on artificial intelligence.

- Data can be noisy which will result in inaccurate predictions.
- Incorrect or incomplete information can also lead to faulty programming through Machine Learning.

2) Fault In Credit Card Fraud Detection

Although this AI-driven software helps to successfully detect credit card fraud, there are issues in Machine Learning that make the process redundant.

3) Getting Bad Recommendations

Proposal engines are quite regular today. While some might be dependable, others may not appear to provide the necessary results. Machine Learning algorithms tend to only impose what these proposal engines have suggested.

4) Talent Deficit

Albeit numerous individuals are pulled into the ML business, however, there are still not many experts who can take complete control of this innovation.

5) Implementation

Organizations regularly have examination engines working with them when they decide to move up to ML. The usage of fresher ML strategies with existing procedures is a complicated errand.

6) Making The Wrong Assumptions

ML models can’t manage datasets containing missing data points. Thus, highlights that contain a huge part of missing data should be erased.

7) Deficient Infrastructure

ML requires a tremendous amount of data stirring abilities. Inheritance frameworks can’t deal with the responsibility and clasp under tension.

8) Having Algorithms Become Obsolete When Data Grows

ML algorithms will consistently require a lot of data when being trained. Frequently, these ML algorithms will be trained over a specific data index and afterwards used to foresee future data, a cycle which you can only expect with a significant amount of effort.

9) Absence Of Skilled Resources

The other issues in Machine Learning are that deep analytics and ML in their present structures are still new technologies.

10) Customer Segmentation

Let us consider the data of human behaviour by a user during a time for testing and the relevant previous practices. All things considered, an algorithm is necessary to recognize those customers that will change over to the paid form of a product and those that won't.

The lists of supervised learning algorithms in ML are:

- Neural Networks
- Naive Bayesian Model
- Classification
- Support Vector Machines
- Regression
- Random Forest Model

11) Complexity

Although Machine Learning and Artificial Intelligence are booming, a majority of these sectors are still in their experimental phases, actively undergoing a trial and error method.

12) Slow Results

Another one of the most common issues in Machine Learning is the slow-moving program. The Machine Learning Models are highly efficient bearing accurate results but the said results take time to be produced.

13) Maintenance

Requisite results for different actions are bound to change and hence the data needed for the same is different.

14) Concept Drift

This occurs when the target variable changes, resulting in the delivered results being inaccurate. This forces the decay of the models as changes cannot be easily accustomed to or upgraded.

15) Data Bias

This occurs when certain aspects of a data set need more importance than others.

16) High Chances Of Error

Many algorithms will contain biased programming which will lead to biased datasets. It will not deliver the right output and produces irrelevant information.

17) Lack Of Explainability

Machine Learning is often termed a “Black box” as deciphering the outcomes from an algorithm is often complex and sometimes useless.

TOPIC-4 Statistical Learning: Introduction

- Structuring and visualizing data are important aspects of data science, the main challenge lies in the mathematical analysis of the data.
- When the goal is to interpret the model and quantify the uncertainty in the data, this analysis is usually referred to as statistical learning.

There are two major goals for modeling data:

- 1) to accurately predict some future quantity of interest, given some observed data, and
- 2) To discover unusual or interesting patterns in the data.

TOPIC-5 Supervised and Unsupervised Learning:

1. Feature, Response:

- Given an input or feature vector x, one of the main goals of machine learning is to predict response an output or response variable y.
- For example, x could be a digitized signature and y a binary variable that indicates whether the signature is genuine or false.

2. Prediction function:

- Another example is where x represents the weight and smoking habits of an expecting mother and y the birth weight of the baby.
- The data science attempt at this prediction is encoded in a mathematical prediction function g, called the prediction function function, which takes as an input x and outputs a guess g(x) for y.

3. Regression, classification:

- In regression problems, the response variable y can take any real value.
- In contrast, regression when y can only lie in a finite set, say $y \in \{0. . . c - 1\}$, then predicting y is conceptually the same as classifying the input x into one of c categories, and so prediction becomes a classification problem.
- loss function:
- We can measure the accuracy of a prediction by with respect to a given response y by loss function using some Loss(y,y').
- In a regression setting the usual choice is the squared error loss $(y-y')\ 2$.

TOPIC-6 Training and Test Loss:

Given an arbitrary prediction function g, it is typically not possible to compute its risk $\ell(g)$ in (2.1). However, using the training sample $\mathcal{T}$, we can approximate $\ell(g)$ via the empirical (sample average) risk

$$\ell_{\mathcal{T}}(g) = \frac{1}{n}\sum_{i=1}^{n} \text{Loss}(Y_i, g(\boldsymbol{X}_i)), \tag{2.3}$$

which we call the *training loss*. The training loss is thus an unbiased estimator of the risk (the expected loss) for a prediction function g, based on the training data.

To approximate the optimal prediction function g^* (the minimizer of the risk $\ell(g)$) we first select a suitable collection of approximating functions $\mathcal{G}$ and then take our *learner* to be the function in $\mathcal{G}$ that minimizes the training loss; that is,

$$g_{\mathcal{T}}^{\mathcal{G}} = \underset{g\in\mathcal{G}}{\text{argmin}}\, \ell_{\mathcal{T}}(g). \tag{2.4}$$

The prediction accuracy of new pairs of data is measured by the *generalization risk* of the learner. For a *fixed* training set τ it is defined as

$$\ell(g_{\tau}^{\mathcal{G}}) = \mathbb{E}\,\text{Loss}(Y, g_{\tau}^{\mathcal{G}}(\boldsymbol{X})), \tag{2.5}$$

For any outcome τ of the training data, we can estimate the generalization risk without bias by taking the sample average

$$\ell_{\mathcal{T}'}(g_\tau^{\mathcal{G}}) := \frac{1}{n'} \sum_{i=1}^{n'} \text{Loss}(Y_i', g_\tau^{\mathcal{G}}(\boldsymbol{X}_i')), \tag{2.7}$$

where $\{(\boldsymbol{X}_1', Y_1'), \ldots, (\boldsymbol{X}_{n'}', Y_{n'}')\} =: \mathcal{T}'$ is a so-called *test sample*. The test sample is completely separate from $\mathcal{T}$, but is drawn in the same way as $\mathcal{T}$; that is, via independent draws from $f(\boldsymbol{x}, y)$, for some sample size n'. We call the estimator (2.7) the *test loss*. For a random training set $\mathcal{T}$ we can define $\ell_{\mathcal{T}'}(g_{\mathcal{T}}^{\mathcal{G}})$ similarly. It is then crucial to assume that $\mathcal{T}$ is independent of $\mathcal{T}'$. Table 2.1 summarizes the main definitions and notation for supervised learning.

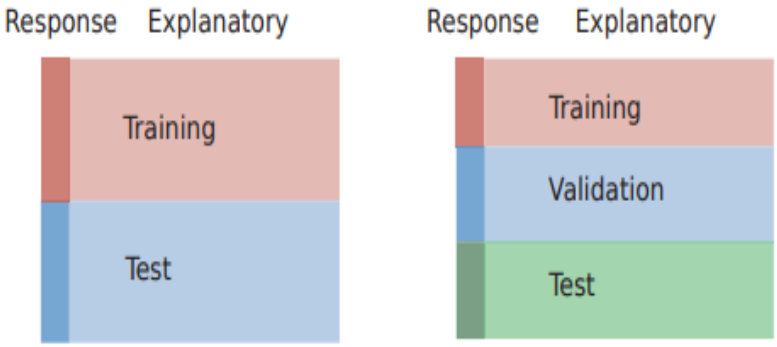


Figure 2.3: Statistical learning algorithms often require the data to be divided into training and test data. If the latter is used for model selection, a third set is needed for testing the performance of the selected model.

TOPIC-7 Tradeoffs in Statistical Learning:

The art of machine learning in the supervised case is to make the generalization risk (2.5) or expected generalization risk (2.6) as small as possible, while using as few computational resources as possible. In pursuing this goal, a suitable class $\mathcal{G}$ of prediction functions has to be chosen. This choice is driven by various factors, such as

- the complexity of the class (e.g., is it rich enough to adequately approximate, or even contain, the optimal prediction function g^*?),
- the ease of training the learner via the optimization program (2.4),
- how accurately the training loss (2.3) estimates the risk (2.1) within class $\mathcal{G}$,
- the feature types (categorical, continuous, etc.).

We can decompose the generalization risk (2.5) into the following three components:

$$\ell(g_\tau^{\mathcal{G}}) = \underbrace{\ell^*}_{\text{irreducible risk}} + \underbrace{\ell(g^{\mathcal{G}}) - \ell^*}_{\text{approximation error}} + \underbrace{\ell(g_\tau^{\mathcal{G}}) - \ell(g^{\mathcal{G}})}_{\text{statistical error}}, \tag{2.16}$$

where $\ell^* := \ell(g^*)$ is the *irreducible risk* and $g^{\mathcal{G}} := \operatorname{argmin}_{g \in \mathcal{G}} \ell(g)$ is the best learner within class $\mathcal{G}$. No learner can predict a new response with a smaller risk than ℓ^*.

The second component is the *approximation error*; it measures the difference between the irreducible risk and the best possible risk that can be obtained by selecting the best prediction function in the selected class of functions $\mathcal{G}$.

The third component is the *statistical (estimation) error*. It depends on the training set τ and, in particular, on how well the learner $g_\tau^{\mathcal{G}}$ estimates the best possible prediction function, $g^{\mathcal{G}}$, within class $\mathcal{G}$. For any sensible estimator this error should decay to zero (in

TOPIC-8 Estimating Risk:

The most straightforward way to quantify the generalization risk (2.5) is to estimate it via the test loss (2.7). However, the generalization risk depends inherently on the training set, and so different training sets may yield significantly different estimates.

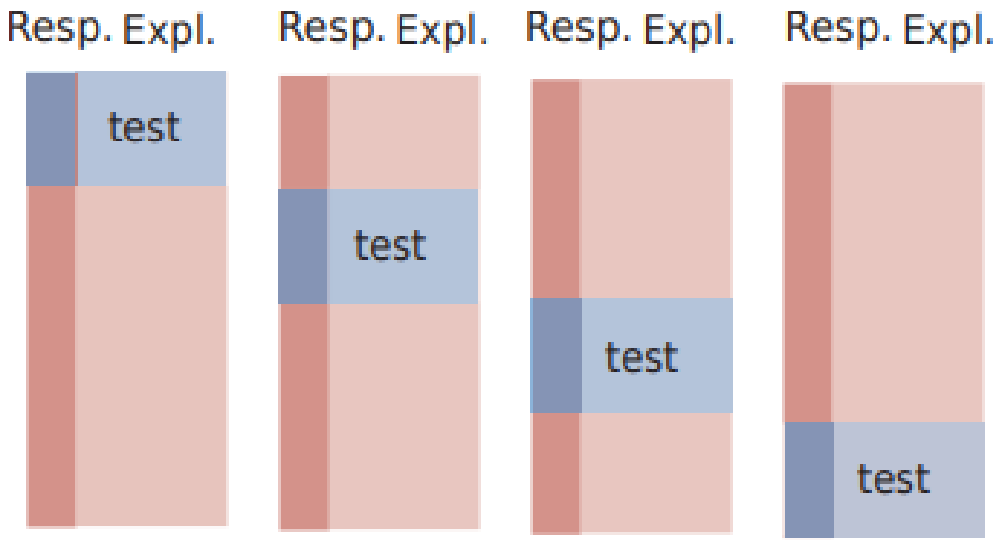
1. IN-SAMPLE RISK:

We mentioned that, due to the phenomenon of overfitting, the training loss of the learner, $\ell_\tau(g_\tau)$ (for simplicity, here we omit $\mathcal{G}$ from $g_\tau^{\mathcal{G}}$), is not a good estimate of the generalization risk $\ell(g_\tau)$ of the learner. One reason for this is that we use the same data for both training the model and assessing its risk. How should we then estimate the generalization risk or expected generalization risk?

To simplify the analysis, suppose that we wish to estimate the average accuracy of the predictions of the learner g_τ at the n feature vectors $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ (these are part of the training set τ). In other words, we wish to estimate the *in-sample risk* of the learner g_τ:

$$\ell_{\text{in}}(g_\tau) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{E}\,\text{Loss}(Y_i', g_\tau(\boldsymbol{x}_i)), \tag{2.23}$$

2. CROSS-VALIDATION



TOPIC-9 Sampling distributions of estimators

Since our estimators are statistics (particular functions of random variables), their distribution can be derived from the joint distribution of X1 . . . Xn.

It is called the sampling distribution because it is based on the joint distribution of the random sample.

-Given a sampling distribution, we can – calculate the probability that an estimator will not differ from the parameter θ by more than a specified amount

– obtain interval estimates rather than point estimates after we have a sample

- An interval estimate is a random interval such that the true parameter lies within this interval with a given probability (say 95%).

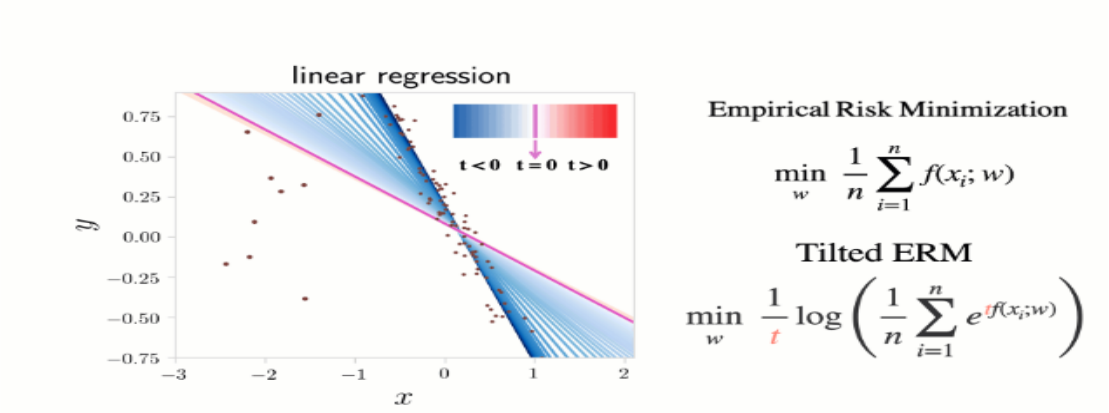
– Choose between to estimators- we can, for instance, calculate the mean-squared error of the estimator, $E\theta[(\theta\hat{} - \theta)\ 2\ ]$ using the distribution of θˆ.

Sampling distributions of estimators depend on sample size, and we want to know exactly how the distribution changes as we change this size so that we can make the right trade-offs between cost and accuracy.

TOPIC-10 Empirical Risk Minimization:

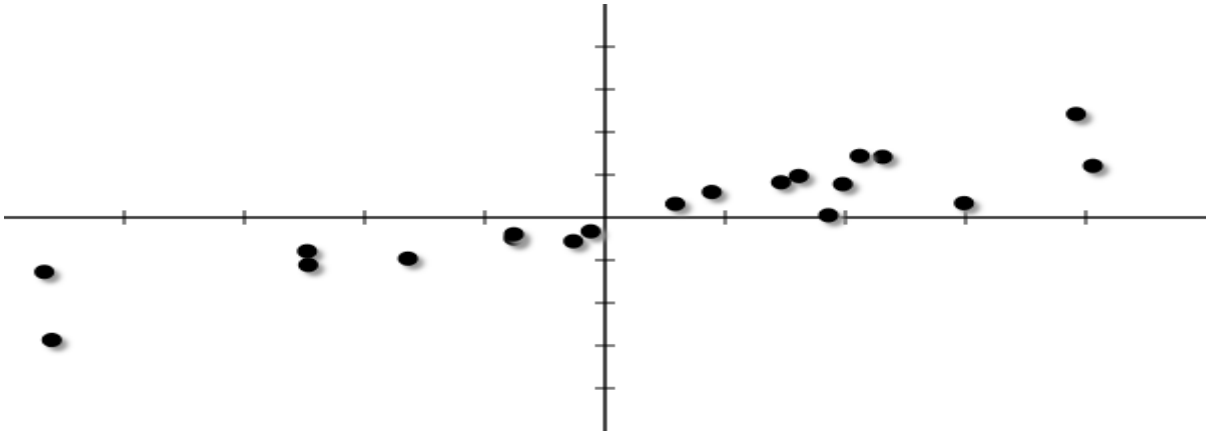
- Empirical Risk Minimization is a fundamental concept in machine learning, yet surprisingly many practitioners are not familiar with it.
- Understanding ERM is essential to understanding the limits of machine learning algorithms and to form a good basis for practical problem-solving skills.
- The theory behind ERM is the theory that explains the VC-dimension, Probably Approximately Correct (PAC) Learning and other fundamental concepts.

Tilted Empirical Risk Minimization



The ERM is a nice idea, if used with care

The plot below shows a regression problem with a training set of 15 points.



The ERM principle is an inference principle which consists in finding the model f^ by minimizing the empirical risk:

$$f\hat{} = \arg\min_{f:X\to Y} R_{emp}(h)$$

where the empirical risk is an estimate of the risk computed as the average of the loss function over the training sample $D=\{(X_i,Y_i)\}_{i=1}^{N}$:

$$R_{emp}(f)=\frac{1}{N}\sum_{i=1}^{N}\ell(f(X_i),Y_i)$$

with the loss function ℓ.