ARTIFICIAL INTELLIGENCE (203105322)

Dr. Pooja Sapra, Associate Professor

Information Technology

Reasoning in AI

The reasoning is the mental process of deriving logical conclusion and making predictions from available knowledge, facts, and beliefs. Or we can say, "Reasoning is a way to infer facts from existing data." It is a general process of thinking rationally, to find valid conclusions.

In artificial intelligence, the reasoning is essential so that the machine can also think rationally as a human brain, and can perform like a human.

Types of Reasoning

- Deductive reasoning
- Inductive reasoning
- Abductive reasoning
- Common Sense Reasoning
- Monotonic Reasoning
- Non-monotonic Reasoning

Deductive reasoning:

Deductive reasoning is deducing new information from logically related known information. It is the form of valid reasoning, which means the argument's conclusion must be true when the premises are true.

Deductive reasoning is a type of propositional logic in AI, and it requires various rules and facts. It is sometimes referred to as top-down reasoning, and contradictory to inductive reasoning.

In deductive reasoning, the truth of the premises guarantees the truth of the conclusion.

Deductive reasoning mostly starts from the general premises to the specific conclusion, which can be explained as below example.

Example:

Premise-1: All the human eats veggies

Premise-2: Suresh is human.

Conclusion: Suresh eats veggies.

Inductive Reasoning:

Inductive reasoning is a form of reasoning to arrive at a conclusion using limited sets of facts by the process of generalization. It starts with the series of specific facts or data and reaches to a general statement or conclusion.

Inductive reasoning is a type of propositional logic, which is also known as cause-effect reasoning or bottom-up reasoning.

In inductive reasoning, we use historical data or various premises to generate a generic rule, for which premises support the conclusion.

In inductive reasoning, premises provide probable supports to the conclusion, so the truth of premises does not guarantee the truth of the conclusion.
Example:
Premise: All of the pigeons we have seen in the zoo are white.

Conclusion: Therefore, we can expect all the pigeons to be white.

Abductive reasoning:

Abductive reasoning is a form of logical reasoning which starts with single or multiple observations then seeks to find the most likely explanation or conclusion for the observation.

Abductive reasoning is an extension of deductive reasoning, but in abductive reasoning, the premises do not guarantee the conclusion.

Example:

Implication: Cricket ground is wet if it is raining

Axiom: Cricket ground is wet.

Conclusion It is raining.

Common Sense Reasoning

Common sense reasoning is an informal form of reasoning, which can be gained through experiences.

Common Sense reasoning simulates the human ability to make presumptions about events which occurs on every day.

It relies on good judgment rather than exact logic and operates on heuristic knowledge and heuristic rules.
Example:

One person can be at one place at a time.
If I put my hand in a fire, then it will burn.
The above two statements are the examples of common sense reasoning which a human mind can easily understand and assume.

Monotonic Reasoning:

In monotonic reasoning, once the conclusion is taken, then it will remain the same even if we add some other information to existing information in our knowledge base. In monotonic reasoning, adding knowledge does not decrease the set of prepositions that can be derived.

To solve monotonic problems, we can derive the valid conclusion from the available facts only, and it will not be affected by new facts.

Monotonic reasoning is not useful for the real-time systems, as in real time, facts get changed, so we cannot use monotonic reasoning.

Monotonic reasoning is used in conventional reasoning systems, and a logic-based system is monotonic.

Any theorem proving is an example of monotonic reasoning.

Example:

Earth revolves around the Sun.

Monotonic Reasoning:

Advantages:

1. In monotonic reasoning, each old proof will always remain valid.
2. If we deduce some facts from available facts, then it will remain valid for always.

Disadvantages:

1. We cannot represent the real-world scenarios using Monotonic reasoning.
2. Hypothesis knowledge cannot be expressed with monotonic reasoning, which means facts should be true.
3. Since we can only derive conclusions from the old proofs, so new knowledge from the real world cannot be added.

Non-monotonic Reasoning

In Non-monotonic reasoning, some conclusions may be invalidated if we add some more information to our knowledge base.

Logic will be said as non-monotonic if some conclusions can be invalidated by adding more knowledge into our knowledge base.

Non-monotonic reasoning deals with incomplete and uncertain models.

"Human perceptions for various things in daily life, "is a general example of non-monotonic reasoning.

Example: Let suppose the knowledge base contains the following knowledge:

Birds can fly

Penguins cannot fly

Pitty is a bird

So from the above sentences, we can conclude that Pitty can fly.

However, if we add one another sentence into knowledge base "Pitty is a penguin", which concludes "Pitty cannot fly", so it invalidates the above conclusion.

Non-Monotonic Reasoning

Advantages of Non-monotonic reasoning:

For real-world systems such as Robot navigation, we can use non-monotonic reasoning.

In Non-monotonic reasoning, we can choose probabilistic facts or can make assumptions.

Disadvantages of Non-monotonic Reasoning:

In non-monotonic reasoning, the old facts may be invalidated by adding new sentences.

It cannot be used for theorem proving.

Forward vs. Backward Reasoning

Forward Chaining the Inference Engine goes through all the facts, conditions and derivations before deducing the outcome i.e When based on available data a decision is taken then the process is called as Forwarding chaining, It works from an initial state and reaches to the goal(final decision).

Example:
A
A -> B
B

He is running.
If he is running, he sweats.
He is sweating.

Parul® University

DIGITAL LEARNING CONTENT

Forward vs. Backward Reasoning

In Backward Reasoning, the inference system knows the final decision or goal, this system starts from the goal and works backwards to determine what facts must be asserted so that the goal can be achieved, i.e it works from goal(final decision) and reaches the initial state.

Example:
B
A -> B
A

He is sweating.

Parul® University

DIGITAL LEARNING CONTENT

Forward vs. Backward Reasoning

1.	When based on available data a decision is taken then the process is called as Forward chaining.	Backward chaining starts from the goal and works backward to determine what facts must be asserted so that the goal can be achieved.
2.	Forward chaining is known as data-driven technique because we reaches to the goal using the available data.	Backward chaining is known as goal-driven technique because we start from the goal and reaches the initial state in order to extract the facts.
3.	It is a bottom-up approach.	It is a top-down approach.
4.	It applies the Breadth-First Strategy.	It applies the Depth-First Strategy.
5.	Its goal is to get the conclusion.	Its goal is to get the possible facts or the required data.
6.	Slow as it has to use all the rules.	Fast as it has to use only a few rules.
7.	It operates in forward direction i.e it works from initial state to final decision.	It operates in backward direction i.e it works from goal to reach initial state.
8.	Forward chaining is used for the planning, monitoring, control, and interpretation application.	It is used in automated inference engines, theorem proofs, proof assistants and other artificial intelligence applications.

Conflict Resolution

If there is a new situation (state) generates, then multiple production rules will be fired together, this is called conflict set. In this situation, the agent needs to select a rule from these sets, and it is called a conflict resolution.

Strategies for Conflict Resolution:

1. Specificity - If all of the conditions of two or more rules are satisfied, choose the rule according to how specific its conditions are.
2. Recency - When two or more rules could be chosen, favor the one that matches the most recently added facts.
3. Not previously used - If a rule's conditions are satisfied, but previously the same rule has been satisfied by the same facts, ignore the rule. This helps to prevent the system from entering infinite loops.
4. Order - Pick the first applicable rule in order of presentation.
5. Arbitrary choice - Pick a rule at random. This has the merit of being simple to compute.

Default Reasoning

It is a very common form of non-monotonic reasoning, where conclusions are drawn based on what is most likely to be true. Two approaches to deal with Default reasoning are as follows:

1. Non-Monotonic Logic: Logic is built to allow statements to be retracted because truth of statements may change when new information is added.

   To represent this uncertainty a modal operator, M is used, which means consistent with everything we know.

   For example: for all (x):playsinstrument(x) and Mmanages(x)->musician(x)

2. Default Logic: It initiates a new inference rule

$$\frac{A:B}{C}$$

Where, A is prerequisite, B is justification and C is consequent

If A is true and is consistent with rets of what is known to assume that B, then conclude C.

Truth Maintenance System

What is TMS?

- A TMS deals with uncertainty by permitting new knowledge to replace old knowledge which is believed to be outdated or erroneous.
- A Truth Maintenance System (TMS) is a Problem Solver module responsible for:
  - ✓ Enforcing logical relations among beliefs.
  - ✓ Generating explanations for conclusions.
  - ✓ Finding solutions to search problems
  - ✓ Supporting default reasoning.
  - ✓ Identifying causes for failure and recover from inconsistencies.

Enforcement of Logical Relationships

1. Enforcement of logical relations

- AI problem -> search.
- Search utilizes assumptions.
- Assumptions change.
- Changing assumptions -> updating consequences of beliefs.
- TMS: mechanism to maintain and update relations among beliefs.

(Every AI problem which is not completely specified requires search. Search utilizes assumptions, which may eventually change. Changing assumptions requires updating consequences of beliefs. Re-derivation of those consequences is most often not desirable, therefore we need a mechanism to maintain and update relations among beliefs.)

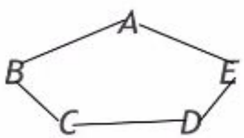
Generations of Explanations

2. Generation of explanations

- Solving problems is what Problem Solvers do.
  - However, often solutions are not enough.
- The PS is expected to provide an explanation
- TMS uses cached inferences for that aim.
- TMS is efficient:
  - Generating cached inferences once is more beneficial than running inference rules that have generated these inferences more than once.

Finding Solutions to Search Problems

3. Finding solutions to search problems



- *Color the nodes: red (1), green (2) yellow (3).*
- *Adjacent nodes are of different colors.*
- *The set of constraints describe this problem:*

A1 or A2 or A3	*not (A1 and B1)*	*not (A3 and C3)*	*not (D2 and E2)*
B1 or B2 or B3	*not (A2 and B2)*	*not (B1 and D1)*	*not (D3 and E3)*
C1 or C2 or C2	*not (A3 and B3)*	*not (B2 and D2)*	*not (C1 and E1)*
D1 or D2 or D3	*not (A1 and C1)*	*not (B3 and D3)*	*not (C2 and E2)*
E1 or E2 or E2	*not (A2 and C2)*	*not (D1 and E1)*	*not (C3 and E3)*

(Assume you want to color the nodes so that every node is red, or green, or yellow, and adjacent nodes are of different colors. Let "1" means "red", "2" means "green", and "3" means "yellow". Then, the following set of constraints describe this problem☺

Default Reasoning and TMS

4. Default reasoning and TMS

Many real-world problems cannot be completely specified. That is, the PS must make conclusions based on incomplete information. Typically the assumption under which such conclusions are drawn is that X is true unless there is an evidence to the contrary. This is known as the "Closed-World Assumption" (CWA). Notice that the CWA helps us limit the underlying search space by assuming only a certain choice and ignoring the others. The reasoning scheme that utilizes this assumption is called "default (or non-monotonic) reasoning".

Example: *Consider the following knowledge base*

Bird(tom) and not Abnormal(tom) → Can_fly(tom)

Penguin(tom) → Abnormal(tom)

Ostrich(tom) → Abnormal(tom)

Bird(tom)

Under the CWA, we assume *not Abnormal(tom)* (because there is no evidence that Tom is abnormal). Therefore, we can derive *can_fly(tom).*

Non-monotonic TMS supports this type of reasoning.

Identifying Causes for failures

5. Identifying causes for failures and recovering from inconsistencies

- Inconsistencies among beliefs in the KB are always possible:
  - wrong data (example: “Outside temperature is 320 degrees.”)
  - Impossible constraints (example: Big-house and Cheap-house and Nice-house).
- TMS maintains help identify the reason for an inconsistency
- Non-monotonic inference. PS is forced to “jump” to a conclusion, because of the lack of information, or lack of time to derive the conclusion.
- “dependency-directed backtracking” allows the TMS to recover.

Uncertainty in Reasoning

In Logic, A→B, means if A is true then B is true,

but consider a situation where we are not sure about whether A is true or not then we cannot express this statement, this situation is called uncertainty.

So, to represent uncertain knowledge, where we are not sure about the predicates, we need uncertain reasoning or probabilistic reasoning.

Causes of uncertainty:

Following are some leading causes of uncertainty to occur in the real world.

1. Information occurred from unreliable sources.
2. Experimental Errors
3. Equipment fault
4. Temperature variation
5. Climate change.

Probabilistic Reasoning

Probabilistic reasoning is a way of knowledge representation where we apply the concept of probability to indicate the uncertainty in knowledge. In probabilistic reasoning, we combine probability theory with logic to handle the uncertainty.

Need of probabilistic reasoning in AI:

- When there are unpredictable outcomes.
- When specifications or possibilities of predicates becomes too large to handle.
- When an unknown error occurs during an experiment.

In probabilistic reasoning, there are two ways to solve problems with uncertain knowledge:

- **Bayes' rule**
- **Bayesian Statistics**

Basics of Probability

Probability: Probability can be defined as a chance that an uncertain event will occur. It is the numerical measure of the likelihood that an event will occur. The value of probability always remains between 0 and 1 that represent ideal uncertainties.

1. $0 \le P(A) \le 1$, where P(A) is the probability of an event A.
2. $P(A) = 0$, indicates total uncertainty in an event A.
3. $P(A) = 1$, indicates total certainty in an event A.

$$\text{Probability of occurrence} = \frac{\text{Number of desired outcomes}}{\text{Total number of outcomes}}$$

Conditional Probability

Conditional probability is a probability of occurring an event when another event has already happened.
Let's suppose, we want to calculate the event A when event B has already occurred, "the probability of A under the conditions of B", it can be written as:

$$P(A|B) = \frac{P(A \wedge B)}{P(B)}$$

Where P($A \wedge B$)= Joint probability of a and B
P(B)= Marginal probability of B.

If the probability of A is given and we need to find the probability of B, then it will be given as:

$$P(B|A) = \frac{P(A \wedge B)}{P(A)}$$

Example

In a class, there are 70% of the students who like English and 40% of the students who likes English and mathematics, and then what is the percent of students those who like English also like mathematics?

Solution:

Let, A is an event that a student likes Mathematics

B is an event that a student likes English.

$$P(A|B) = \frac{P(A \wedge B)}{P(B)} = \frac{0.4}{0.7} = 57\%$$

Bayes' Theorem

It is a way to calculate the value of P(B|A) with the knowledge of P(A|B).

Bayes' theorem allows updating the probability prediction of an event by observing new information of the real world.

Example: If cancer corresponds to one's age then by using Bayes' theorem, we can determine the probability of cancer more accurately with the help of age.

Bayes' theorem can be derived using product rule and conditional probability of event A with known event B:

As from product rule we can write:

1.$P(A \wedge B) = P(A|B)\, P(B)$ or

Similarly, the probability of event B with known event A:

2.$P(A \wedge B) = P(B|A)\, P(A)$

Equating right hand side of both the equations, we will get:

$$P(A|B) = \frac{P(B|A)\, P(A)}{P(B)} \quad \text{....(a)}$$

Bayes' rule or **Bayes' theorem**

Example- Bayes' Theorem

A doctor is aware that disease meningitis causes a patient to have a stiff neck, and it occurs 80% of the time. He is also aware of some more facts, which are given as follows:

- The Known probability that a patient has meningitis disease is 1/30,000.
- The Known probability that a patient has a stiff neck is 2%.

Let a be the proposition that patient has stiff neck and b be the proposition that patient has meningitis. , so we can calculate the following as:

$P(a|b) = 0.8$

$P(b) = 1/30000$

$P(a) = .02$

$$P(b|a) = \frac{P(a|b)P(b)}{P(a)} = \frac{0.8 * (\frac{1}{30000})}{0.02} = 0.001333333.$$

Hence, we can assume that 1 patient out of 750 patients has meningitis disease with a stiff neck.

Bayesian Belief Network in artificial intelligence

"A Bayesian network is a probabilistic graphical model which represents a set of variables and their conditional dependencies using a directed acyclic graph."

It is also called a **Bayes network, belief network, decision network**, or **Bayesian model**.

Bayesian networks are probabilistic, because these networks are built from a **probability distribution**, and also use probability theory for prediction and anomaly detection.

Real world applications are probabilistic in nature, and to represent the relationship between multiple events, we need a Bayesian network. It can also be used in various tasks including **prediction, anomaly detection, diagnostics, automated insight, reasoning, time series prediction**, and **decision making under uncertainty**.

Bayesian Network can be used for building models from data and experts opinions, and it consists of two parts:

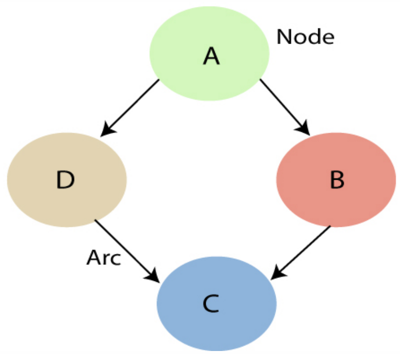
- **Directed Acyclic Graph**
- **Table of conditional probabilities.**

Bayesian Belief Network in artificial intelligence

A Bayesian network graph is made up of nodes and Arcs (directed links), where:

- Each **node** corresponds to the random variables, and a variable can be **continuous** or **discrete**.
- **Arc or directed arrows** represent the causal relationship or conditional probabilities between random variables. These directed links or arrows connect the pair of nodes in the graph.

These links represent that one node directly influence the other node, and if there is no directed link that means that nodes are independent with each other



- **In the above diagram, A, B, C, and D are random variables represented by the nodes of the network graph.**
- **If we are considering node B, which is connected with node A by a directed arrow, then node A is called the parent of Node B.**
- **Node C is independent of node A.**

Bayesian Belief Network in artificial intelligence

Bayesian network is based on Joint probability distribution and conditional probability.

If we have variables x1, x2, x3,....., xn, then the probabilities of a different combination of x1, x2, x3.. xn, are known as Joint probability distribution.

$\mathbf{P[x_1, x_2, x_3,\ldots\ldots, x_n]}$, it can be written as the following way in terms of the joint probability distribution.

$$= \mathbf{P[x_1| x_2, x_3,\ldots\ldots, x_n]P[x_2, x_3,\ldots\ldots, x_n]}$$
$$= \mathbf{P[x_1| x_2, x_3,\ldots\ldots, x_n]P[x_2|x_3,\ldots\ldots, x_n]\ldots.P[x_{n-1}|x_n]P[x_n].}$$

In general for each variable Xi, we can write the equation as:

$$P(X_i|X_{i-1},\ldots\ldots\ldots, X_1) = P(X_i\ |Parents(X_i\ ))$$

Bayesian Belief Network in artificial intelligence- Example

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.

List of all events occurring in this network:

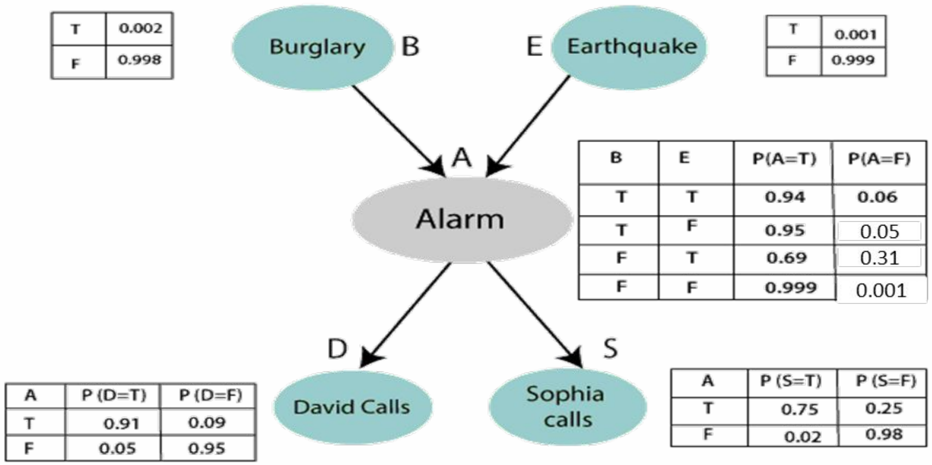
- **Burglary (B)**
- **Earthquake(E)**
- **Alarm(A)**
- **David Calls(D)**
- **Sophia calls(S)**

Bayesian Belief Network in artificial intelligence

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 conditional distributions for each node are given as conditional probabilities table or CPT.

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



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	0.05
F	T	0.69	0.31
F	F	0.999	0.001

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

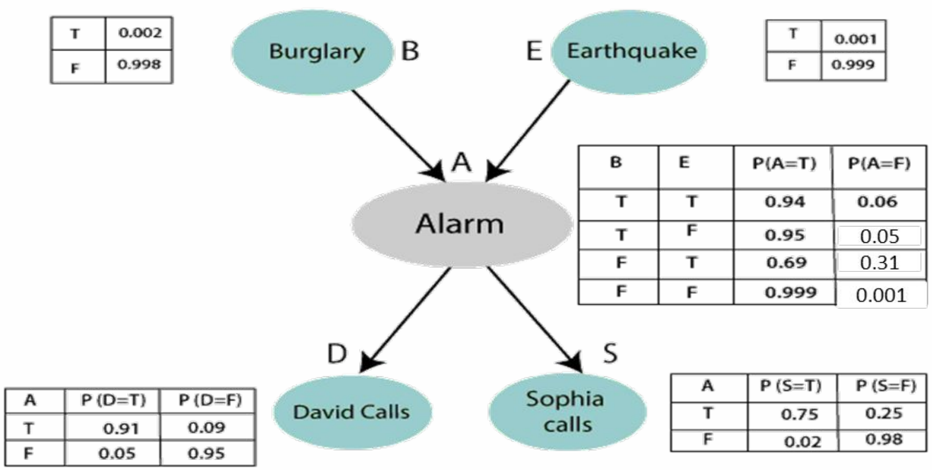
Bayesian Belief Network in artificial intelligence

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.



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	0.05
F	T	0.69	0.31
F	F	0.999	0.001

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

Utility Theory – Making Decisions

For Making Simple Decisions we need reasoning under uncertainty, i.e., methods for representing and reasoning about state information that we do not know with certainty.

Probabilistic model was used for this.

For our agent-based system, we now need to use this information within a process that allows the agent to act rationally.

In other words, we need to determine how to make good decisions over actions based on our beliefs.

- The framework to guide decision making will be based on utility theory.
- An agent's preferences between world states will be represented via a utility function.
- The utility function will assign a number to each state to express desirability of that state.

- Let U(S) denote the utility of state S.
- Given an action A, let Resulti(A) be the outcome state resulting from applying action A.
- The principle of maximum expected utility (MEU) states that a rational agent should choose an action that maximizes the agent's expected utility.

Markov Models

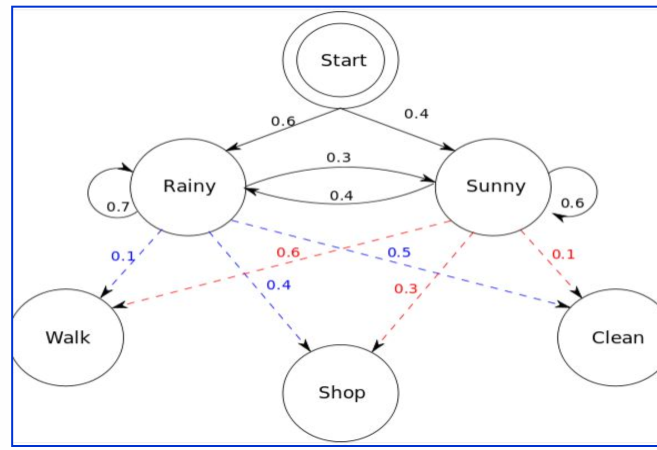
Markov models are a type of probabilistic model that is used to predict the future state of a system, based on its current state. In other words, Markov models are used to predict the future state based on the current states.

HMM is a statistical model in which the system being modeled are Markov processes with unobserved or hidden states.

A hidden Markov model consists of five important components:

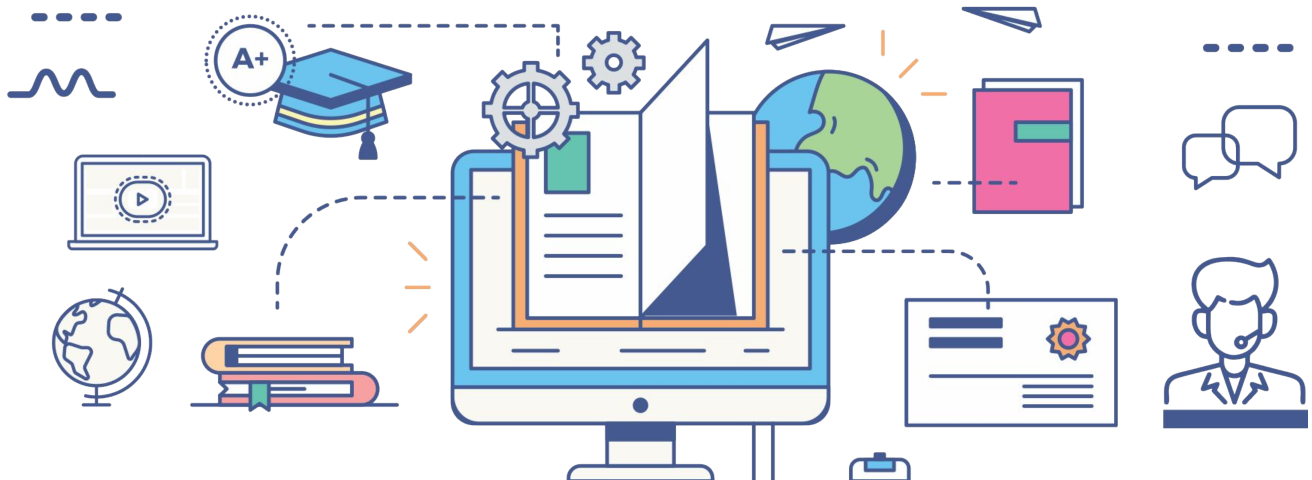
Initial probability distribution
One or more hidden states
Transition probability distribution
A sequence of observations
Emission probabilities

There are two hidden states such as rainy and sunny. These states are hidden because what is observed as the process output is whether the person is shopping, walking, or cleaning.
The sequence of observations is shop, walk, and clean.
An initial probability distribution is represented by start probability
Transition probability represents the transition of one state (rainy or sunny) to another state given the current state
Emission probability represents the probability of observing the output, shop, clean and walk given the states, rainy or sunny.




```
states = ('Rainy', 'Sunny')

observations = ('walk', 'shop', 'clean')

start_probability = {'Rainy': 0.6, 'Sunny': 0.4}

transition_probability = {
   'Rainy' : {'Rainy': 0.7, 'Sunny': 0.3},
   'Sunny' : {'Rainy': 0.4, 'Sunny': 0.6},
   }

emission_probability = {
   'Rainy' : {'walk': 0.1, 'shop': 0.4, 'clean': 0.5},
   'Sunny' : {'walk': 0.6, 'shop': 0.3, 'clean': 0.1},
   }
```

DIGITAL LEARNING CONTENT



Parul® University

www.paruluniversity.ac.in