UNIT 3: PROBABILISTIC REASONING

1. PROBABILITY

1.1 Introduction to Probability

Probability is a mathematical framework used to measure uncertainty. In Artificial Intelligence, many real-world situations involve incomplete or uncertain information. Instead of making decisions with absolute certainty, AI systems often rely on probability to estimate how likely an event is to occur. Probability helps intelligent systems make rational decisions even when full information is not available.

In simple terms, probability tells us how likely something is to happen. Its value ranges between 0 and 1. A probability of 0 means the event will never happen, and a probability of 1 means the event will definitely happen.

1.2 Basic Concepts in Probability

To understand probability properly, it is important to know some basic terms that are frequently used in probability theory and Artificial Intelligence.

Important basic concepts include:

- Experiment - A process that produces an outcome, such as tossing a coin.
- Outcome - A possible result of an experiment.
- Sample Space - The set of all possible outcomes.
- Event - A subset of the sample space.

For example, when tossing a coin, the sample space is {Head, Tail}. If we are interested in getting a head, then the event is {Head}.

1.3 Mathematical Definition of Probability

If all outcomes in a sample space are equally likely, the probability of an event A is defined as:

$P(A)$ = (Number of favorable outcomes) / (Total number of possible outcomes)

For example, when rolling a fair six-sided die, the probability of getting the number 3 is 1/6 because there is one favorable outcome and six total possible outcomes.

1.4 Properties of Probability

Probability follows certain important properties that must always be satisfied.

Important properties include:

- $0 \le P(A) \le 1$
- $P(S) = 1$, where S is the sample space.
- P(Empty Set) = 0
- For mutually exclusive events A and B: $P(A \cup B) = P(A) + P(B)$

These properties ensure that probability values remain logically consistent.

1.5 Types of Probability

There are different interpretations of probability depending on the context.

Common types include:

- Classical Probability - Based on equally likely outcomes.
- Frequentist Probability - Based on long-run frequency of occurrence.
- Subjective Probability - Based on personal belief or judgment.

In Artificial Intelligence, subjective and frequentist interpretations are commonly used because many real-world problems involve uncertainty and incomplete data.

1.6 Importance of Probability in Artificial Intelligence

Probability plays a crucial role in Artificial Intelligence because many AI systems must make decisions under uncertainty. For example, medical diagnosis systems estimate the probability of diseases based on symptoms. Speech recognition systems estimate the probability of words based on sound patterns.

Without probability, AI systems would not be able to handle uncertain or incomplete information effectively. Probability provides a structured way to reason about uncertainty and make rational decisions.

1.7 Example for Better Understanding

Suppose a weather prediction system estimates that there is a 0.8 probability of rain tomorrow. This means there is an 80 percent chance of rain. Based on this probability, a person may decide to carry an umbrella. This simple example shows how probability helps in decision-making under uncertainty.

SUMMARY OF THE TOPIC

Probability is a mathematical measure of uncertainty and plays a vital role in Artificial Intelligence. It helps AI systems make decisions when complete information is not available. Probability values range between 0 and 1 and follow specific mathematical properties. Understanding probability is essential for studying advanced topics such as conditional probability, Bayes rule, and Bayesian networks.

2. CONDITIONAL PROBABILITY

2.1 Introduction to Conditional Probability

Conditional Probability refers to the probability of an event occurring given that another event has already occurred. In many real-world situations, the occurrence of one event affects the likelihood of another event. Conditional probability helps in updating our belief about an event when new information becomes available.

In Artificial Intelligence, conditional probability is extremely important because AI systems often need to revise their predictions based on observed evidence. It forms the foundation for Bayesian reasoning and probabilistic inference.

2.2 Mathematical Definition of Conditional Probability

The conditional probability of an event A given that event B has occurred is denoted as $P(A \mid B)$. It is defined as:

$P(A \mid B) = P(A \cap B) / P(B)$, provided that $P(B) > 0$.

Here, $P(A \cap B)$ represents the probability that both events A and B occur together, and $P(B)$ represents the probability that event B occurs.

2.3 Explanation of the Formula

The formula shows that when event B has already occurred, the sample space is reduced to only those outcomes where B is true. From this reduced sample space, we calculate the probability of A occurring. In simple words, we focus only on situations where B happens and then check how often A also happens in those situations.

2.4 Example for Better Understanding

Suppose a class has 10 students, out of which 6 are boys and 4 are girls. Among the 6 boys, 3 wear glasses. If we randomly select a student and we are told that the student is a boy, what is the probability that he wears glasses?

Here, event A is 'wears glasses' and event B is 'student is a boy'. Since 3 out of 6 boys wear glasses, the conditional probability is:

$P(A \mid B) = 3 / 6 = 1/2$.

This example shows how additional information changes the probability calculation.

2.5 Multiplication Rule of Probability

Conditional probability leads to the multiplication rule, which is used to calculate the joint probability of two events. The multiplication rule states:

$P(A \cap B) = P(A \mid B) \times P(B)$

This rule is widely used in probabilistic reasoning and forms the basis for Bayes' theorem and Bayesian networks.

2.6 Independent and Dependent Events

Two events are said to be independent if the occurrence of one event does not affect the probability of the other event. In this case:

$P(A \mid B) = P(A)$

If the occurrence of one event affects the probability of another event, then the events are dependent. In most AI problems, events are dependent, and conditional probability is required to model their relationships.

2.7 Importance of Conditional Probability in AI

Conditional probability is fundamental in Artificial Intelligence because AI systems must update their beliefs when new evidence is observed. For example, in medical diagnosis, the probability of a disease changes when symptoms are observed. Similarly, in spam filtering, the probability that an email is spam changes when certain keywords are detected.

This ability to update beliefs based on evidence is known as probabilistic inference, which is a core concept in modern AI systems.

SUMMARY OF THE TOPIC

Conditional Probability measures the probability of an event occurring given that another event has already occurred. It is mathematically defined as $P(A \mid B) = P(A \cap B) / P(B)$. It helps in updating probabilities when new information becomes available. Conditional probability forms the basis for advanced probabilistic reasoning techniques such as Bayes' rule and Bayesian networks. Understanding this concept is essential for modeling uncertainty in Artificial Intelligence.

3. BAYES' RULE

3.1 Introduction to Bayes' Rule

Bayes' Rule, also known as Bayes' Theorem, is one of the most important concepts in probabilistic reasoning and Artificial Intelligence. It provides a mathematical way to update the probability of a hypothesis when new evidence is observed. In simple words, Bayes' Rule helps us revise our belief about something after getting additional information.

In many AI applications such as medical diagnosis, spam detection, and speech recognition, systems continuously update their beliefs based on new data. Bayes' Rule forms the foundation for this type of reasoning under uncertainty.

3.2 Mathematical Formula of Bayes' Rule

Bayes' Rule is derived from the multiplication rule of probability. It is expressed as:

$$P(A \mid B) = [P(B \mid A) \times P(A)] / P(B)$$

Where:

- $P(A \mid B)$ is the posterior probability - the updated probability of A given B.
- $P(B \mid A)$ is the likelihood - the probability of observing B given A.
- $P(A)$ is the prior probability - the initial belief about A before seeing evidence.
- $P(B)$ is the evidence - the total probability of observing B.

This formula shows how prior knowledge is combined with new evidence to produce an updated belief.

3.3 Understanding Prior, Likelihood, and Posterior

To understand Bayes' Rule clearly, it is important to understand three main components: prior, likelihood, and posterior. The prior probability represents our initial belief before observing any evidence. The likelihood represents how likely the evidence is if the hypothesis is true. The posterior probability is the updated belief after considering the evidence.

Bayes' Rule allows us to move from prior probability to posterior probability in a logical and mathematical manner.

3.4 Example for Better Understanding

Suppose a disease affects 1 percent of a population. A medical test is 90 percent accurate, meaning it correctly detects the disease in 90 percent of infected individuals. If a person tests positive, what is the probability that the person actually has the disease?

Let A be the event that a person has the disease and B be the event that the test is positive.

Using Bayes' Rule: $P(A \mid B) = [P(B \mid A) \times P(A)] / P(B)$.

Even though the test is highly accurate, the actual probability of having the disease after a positive test depends on the prior probability of the disease in the population. This example shows how Bayes' Rule adjusts probabilities based on background information.

3.5 Extension to Multiple Hypotheses

In many real-world AI problems, there may be multiple possible hypotheses instead of just one. In such cases, Bayes' Rule can be extended using the formula:

$$P(Hi \mid E) = [P(E \mid Hi) \times P(Hi)] / \Sigma [P(E \mid Hj) \times P(Hj)]$$

Here, Hi represents a particular hypothesis and E represents the evidence. The denominator ensures that the probabilities of all possible hypotheses sum to 1.

3.6 Importance of Bayes' Rule in Artificial Intelligence

Bayes' Rule is fundamental in AI because it allows systems to learn from data and update beliefs. It is widely used in Bayesian networks, Naive Bayes classifiers, Hidden Markov Models, and many other probabilistic models. Without Bayes' Rule, it would be difficult to perform logical reasoning under uncertainty.

It enables AI systems to combine prior knowledge with observed data to make better predictions and decisions.

SUMMARY OF THE TOPIC

Bayes' Rule provides a mathematical method to update probabilities when new evidence is observed. It connects prior probability, likelihood, and posterior probability in a structured way. The formula $P(A \mid B) = [P(B \mid A) \times P(A)] / P(B)$ is widely used in Artificial Intelligence for reasoning under uncertainty. Understanding Bayes' Rule is essential for studying advanced topics such as Bayesian networks and probabilistic inference.

4. BAYESIAN NETWORKS - REPRESENTATION

4.1 Introduction to Bayesian Networks

A Bayesian Network is a graphical model used in Artificial Intelligence to represent probabilistic relationships among a set of variables. It provides a compact and structured way to represent joint probability distributions. Bayesian Networks are also known as Belief Networks or Probabilistic Directed Graphical Models.

Instead of listing all possible probability combinations in a large table, a Bayesian Network uses a graph structure to show dependencies between variables. This makes reasoning under uncertainty more efficient and understandable.

4.2 Structure of a Bayesian Network

A Bayesian Network consists of two main components: a directed acyclic graph (DAG) and conditional probability tables (CPTs). The graph represents the structure of dependencies, while the CPTs quantify the strength of those dependencies.

The structural elements include:

- Nodes - Represent random variables.
- Directed Edges - Represent causal or dependency relationships.
- No Cycles - The graph must be acyclic.

If there is a directed edge from variable A to variable B, it means that A has a direct influence on B. A is called the parent of B, and B is called the child of A.

4.3 Directed Acyclic Graph (DAG)

The structure of a Bayesian Network is always a Directed Acyclic Graph. Directed means that the edges have a direction, and acyclic means that there are no loops. This ensures that the relationships between variables are logically consistent and do not form circular dependencies.

The DAG encodes conditional independence relationships between variables. If two variables are not connected directly or indirectly through parents, they may be conditionally independent.

4.4 Conditional Probability Table (CPT)

Each node in a Bayesian Network has an associated Conditional Probability Table. The CPT specifies the probability of the variable given its parent variables.

If a node has no parents, its CPT simply contains its prior probabilities. If a node has parents, the CPT lists probabilities for all possible combinations of parent values.

For example, if variable B has parent A, then the CPT will contain values for $P(B \mid A)$.

4.5 Joint Probability Representation

One of the most powerful properties of Bayesian Networks is that they allow the joint probability distribution of all variables to be expressed as a product of conditional probabilities.

If the network contains variables X1, X2, X3, ..., Xn, then the joint probability can be written as:

$P(X1, X2, ..., Xn) = \Pi\, P(Xi \mid Parents(Xi))$

This factorization greatly reduces the number of probabilities that need to be stored compared to a full joint probability table.

4.6 Example for Better Understanding

Consider a simple example involving three variables: Rain, Sprinkler, and Wet Grass. Rain and Sprinkler both influence whether the grass becomes wet. In this network, Rain and Sprinkler are parent nodes, and Wet Grass is the child node.

The network structure represents how these variables are related, and the CPT for Wet Grass specifies the probability of grass being wet given different combinations of Rain and Sprinkler. This structured representation is much simpler than listing all possible probability combinations in a single large table.

4.7 Advantages of Bayesian Network Representation

Bayesian Networks offer several advantages in representing probabilistic knowledge.

Major advantages include:

- Compact representation of joint probability distribution.
- Clear visualization of dependencies between variables.
- Efficient reasoning under uncertainty.
- Ability to encode causal relationships.

These advantages make Bayesian Networks widely used in diagnosis systems, prediction models, and decision support systems.

SUMMARY OF THE TOPIC

Bayesian Networks are graphical models used to represent probabilistic relationships among variables. They consist of a Directed Acyclic Graph and Conditional Probability Tables. The graph shows dependencies, while CPTs quantify them. Bayesian Networks allow compact representation of joint probability distributions and support efficient reasoning under uncertainty. Understanding their representation is essential for studying construction and inference in probabilistic models.

5. BAYESIAN NETWORKS - CONSTRUCTION

5.1 Introduction to Construction of Bayesian Networks

Construction of a Bayesian Network refers to the process of building its structure and defining the associated probability values. A well-constructed Bayesian Network accurately represents the dependencies among variables and allows efficient probabilistic reasoning. The construction process involves identifying relevant variables, determining their relationships, and assigning conditional probability values.

In Artificial Intelligence, Bayesian Networks can be constructed either manually using expert knowledge or automatically using data-driven learning techniques.

5.2 Steps in Constructing a Bayesian Network

The construction of a Bayesian Network generally follows a systematic process. Each step ensures that the model accurately represents the problem domain.

The main steps include:

- Identify the set of relevant random variables.
- Determine the possible values for each variable.
- Define the directed edges representing dependencies.
- Ensure the graph remains acyclic.
- Construct Conditional Probability Tables (CPTs) for each node.

These steps collectively form the structure and quantitative representation of the network.

5.3 Choosing Variables

The first step in construction is selecting appropriate variables that describe the problem. These variables should represent important aspects of the system being modeled. For example, in a medical diagnosis system, variables may include symptoms, diseases, test results, and patient history.

Choosing irrelevant or excessive variables may increase complexity without improving accuracy. Therefore, careful selection of variables is important.

5.4 Defining Structure and Dependencies

After selecting variables, the next step is defining the relationships between them. Directed edges are added to represent causal or dependency relationships. If one variable directly influences another, an edge is drawn from the cause to the effect.

The structure should reflect conditional independence assumptions. For example, if two variables are independent given a third variable, the structure should represent this relationship clearly.

5.5 Constructing Conditional Probability Tables (CPTs)

Once the structure is defined, Conditional Probability Tables must be assigned to each node. For nodes without parents, prior probabilities are specified. For nodes with parents, probabilities are specified for all combinations of parent values.

These probability values can be obtained from expert knowledge, historical data, or statistical estimation techniques.

5.6 Construction Using Expert Knowledge vs Data

Bayesian Networks can be constructed using domain expert knowledge or learned from data. Expert-based construction relies on human understanding of relationships between variables. Data-driven construction uses machine learning algorithms to determine structure and probabilities.

In many real-world systems, a combination of expert knowledge and data learning is used to improve accuracy and reliability.

5.7 Challenges in Constructing Bayesian Networks

Constructing a Bayesian Network can be challenging, especially when dealing with large numbers of variables. Determining correct dependencies and estimating accurate probabilities requires careful analysis.

Common challenges include:

- Large number of variables.
- Insufficient or incomplete data.
- Complex dependency relationships.
- Ensuring the graph remains acyclic.

Despite these challenges, Bayesian Networks remain powerful tools for modeling uncertainty.

SUMMARY OF THE TOPIC

Construction of a Bayesian Network involves selecting variables, defining dependencies, and assigning conditional probabilities. The process may use expert knowledge, data-driven methods, or a combination of both. A properly constructed network provides a compact and accurate representation of probabilistic relationships. Understanding the construction process is essential before performing inference in Bayesian Networks.

6. BAYESIAN NETWORKS - INFERENCE

6.1 Introduction to Inference in Bayesian Networks

Inference in Bayesian Networks refers to the process of computing probabilities of certain variables given some observed evidence. In simple words, inference means answering questions such as: 'What is the probability of a disease given certain symptoms?' Bayesian inference allows us to update our beliefs about unknown variables when new information becomes available.

Inference is the main purpose of constructing Bayesian Networks. Once the structure and conditional probability tables are defined, the network can be used to calculate posterior probabilities efficiently.

6.2 Types of Inference in Bayesian Networks

There are different types of inference queries that can be performed in a Bayesian Network depending on the type of question being asked.

Common types include:

- Predictive Inference - Predicting effects from known causes.
- Diagnostic Inference - Inferring causes from observed effects.
- Intercausal Inference - Reasoning between causes of a common effect.
- Mixed Inference - Combination of predictive and diagnostic reasoning.

These types of inference allow flexible reasoning in complex probabilistic systems.

6.3 Exact Inference Methods

Exact inference computes the exact posterior probabilities using mathematical operations. The most common exact inference methods include Enumeration and Variable Elimination.

Enumeration Method:

Enumeration calculates probabilities by summing over all possible combinations of hidden variables. Although it gives exact results, it can be computationally expensive for large networks.

Variable Elimination Method:

Variable Elimination improves efficiency by eliminating hidden variables step by step instead of enumerating all combinations. It reduces unnecessary computations and is more efficient than basic enumeration.

6.4 Approximate Inference Methods

In large and complex networks, exact inference may become computationally expensive. In such cases, approximate inference techniques are used to estimate probabilities.

Common approximate methods include:

- Sampling Methods (such as Monte Carlo sampling).
- Rejection Sampling.
- Likelihood Weighting.
- Gibbs Sampling.

These methods provide approximate results but are more practical for large-scale real-world problems.

6.5 Steps in Performing Inference

The general process of performing inference in a Bayesian Network follows a systematic approach.

The steps include:

- Identify the query variable.
- Identify the observed evidence variables.
- Apply probability rules and network structure.
- Compute posterior probability.

The result is a probability value that represents the updated belief after considering the evidence.

6.6 Example for Better Understanding

Consider a Bayesian Network with variables Rain, Sprinkler, and Wet Grass. Suppose we observe that the grass is wet. Using inference, we can compute the probability that it rained given that the grass is wet. This is a diagnostic inference problem. The network structure and CPT values help in calculating the updated probability.

6.7 Importance of Inference in Artificial Intelligence

Inference is essential in AI because it allows intelligent systems to make decisions under uncertainty. Applications include medical diagnosis, fault detection, risk analysis, natural language processing, and robotics. Bayesian inference provides a mathematically sound framework for updating beliefs based on evidence.

SUMMARY OF THE TOPIC

Inference in Bayesian Networks is the process of computing posterior probabilities given evidence. It allows AI systems to update beliefs logically and mathematically. There are exact methods such as Enumeration and Variable Elimination, and approximate methods such as Sampling techniques. Inference enables intelligent reasoning under uncertainty and is one of the most important applications of Bayesian Networks in Artificial Intelligence.

7. TEMPORAL MODEL

7.1 Introduction to Temporal Models

A Temporal Model is a probabilistic model that represents how a system evolves over time. In many real-world problems, the current state of a system depends not only on present conditions but also on previous states. Temporal models capture this time-dependent behavior and allow reasoning about sequences of events.

In Artificial Intelligence, temporal models are used in applications such as speech recognition, weather prediction, stock market analysis, robotics, and tracking systems. These models help in predicting future states based on past and present information.

7.2 Need for Temporal Models

Traditional Bayesian Networks represent static relationships between variables at a single point in time. However, many systems are dynamic, meaning their state changes over time. Temporal models extend Bayesian Networks to handle such dynamic environments.

For example, the health condition of a patient today may depend on their health condition yesterday. Similarly, the position of a robot at time t depends on its position at time t-1. Temporal models provide a structured way to represent these time-based dependencies.

7.3 Time Slices and State Transitions

Temporal models divide time into discrete steps called time slices. Each time slice represents the state of the system at a particular time. Variables are replicated across time slices to represent how they evolve.

The relationship between consecutive time slices is modeled using transition probabilities. These probabilities define how the state at time t influences the state at time t+1.

Important concepts include:

- Time Slice - Representation of variables at a specific time.
- State Transition - Change from one state to another over time.
- Transition Model - Defines $P(X_t \mid X_{t-1})$.

This structure allows prediction of future states and analysis of past states.

7.4 Markov Assumption in Temporal Models

Many temporal models are based on the Markov Assumption. This assumption states that the current state depends only on the immediate previous state and not on the entire history. In mathematical terms:

$$P(X_t \mid X_1, X_2, ..., X_{t-1}) = P(X_t \mid X_{t-1})$$

This assumption simplifies computation because the model does not need to consider all past states. It reduces complexity and makes inference more efficient.

7.5 Dynamic Bayesian Networks (DBN)

Dynamic Bayesian Networks are an extension of Bayesian Networks designed to model temporal processes. A DBN represents a sequence of Bayesian Networks, one for each time slice, connected through transition probabilities.

DBNs combine the structure of Bayesian Networks with temporal dependencies. They are widely used in tracking systems, forecasting, and sequential decision-making problems.

7.6 Inference in Temporal Models

Inference in temporal models involves computing probabilities over sequences of states. Common inference tasks include prediction, filtering, smoothing, and most likely sequence estimation.

Important inference tasks include:

- Prediction - Estimating future states.
- Filtering - Estimating current state given past evidence.
- Smoothing - Estimating past state given all evidence.
- Most Likely Sequence - Determining the most probable sequence of states.

These inference methods are essential in real-time systems and sequential data analysis.

7.7 Example for Better Understanding

Consider a weather prediction system where the weather today depends on yesterday’s weather. If yesterday was rainy, there is a higher probability that today will also be rainy. This relationship is modeled using transition probabilities between time slices. Using this model, we can predict future weather conditions based on current observations.

SUMMARY OF THE TOPIC

Temporal Models represent systems that evolve over time. They divide time into slices and use transition probabilities to model changes between states. The Markov Assumption simplifies computation by assuming that the current state depends only on the previous state. Dynamic Bayesian Networks extend Bayesian Networks to handle time-dependent processes. Temporal models are widely used in forecasting, tracking, and sequential decision-making problems.

8. HIDDEN MARKOV MODEL (HMM)

8.1 Introduction to Hidden Markov Model

A Hidden Markov Model (HMM) is a statistical model used to represent systems that evolve over time where the true state of the system is not directly observable. In such models, we can observe some outputs, but the actual underlying states remain hidden. HMM is widely used in Artificial Intelligence for modeling sequential data.

The term 'Hidden' refers to the fact that the internal states are not directly visible, while 'Markov Model' refers to the use of the Markov assumption that the current state depends only on the previous state.

8.2 Components of Hidden Markov Model

A Hidden Markov Model consists of the following main components:

- Hidden States - The actual states of the system that are not directly observable.
- Observations - The visible outputs generated by the hidden states.
- Transition Probabilities - Probabilities of moving from one hidden state to another.
- Emission Probabilities - Probabilities of generating observations from hidden states.
- Initial State Distribution - Probability distribution over initial states.

These components together define the complete structure of an HMM.

8.3 Markov Property in Hidden Markov Model

The Hidden Markov Model is based on the Markov property, which states that the probability of the current state depends only on the previous state. This simplifies computation and reduces the complexity of modeling long sequences.

Mathematically, it is written as:

$$P(X_t \mid X_1, X_2, ..., X_{t-1}) = P(X_t \mid X_{t-1})$$

This assumption makes HMM suitable for sequential problems such as speech recognition and language modeling.

8.4 Three Fundamental Problems of HMM

There are three main computational problems associated with Hidden Markov Models.

1. Evaluation Problem:

Given a sequence of observations and a model, determine the probability of the observation sequence. This is solved using the Forward Algorithm.

2. Decoding Problem:

Given a sequence of observations, determine the most likely sequence of hidden states. This is solved using the Viterbi Algorithm.

3. Learning Problem:

Given a sequence of observations, estimate the model parameters that maximize the probability of the observations. This is solved using the Baum-Welch Algorithm.

8.5 Applications of Hidden Markov Model

Hidden Markov Models are widely used in many real-world applications involving sequential data.

Important applications include:

- Speech Recognition.
- Handwriting Recognition.
- Part-of-Speech Tagging in Natural Language Processing.
- Bioinformatics and Gene Prediction.

- Activity Recognition in Robotics.

These applications demonstrate the practical importance of HMM in Artificial Intelligence.

8.6 Example for Better Understanding

Consider a simple weather example where the hidden states are 'Sunny' and 'Rainy'. We cannot directly observe the weather state, but we observe whether a person carries an umbrella. If the person carries an umbrella, there is a higher probability that it is rainy. Using transition and emission probabilities, we can estimate the most likely sequence of weather states based on observed umbrella usage.

SUMMARY OF THE TOPIC

A Hidden Markov Model is a temporal probabilistic model used to represent systems with hidden states. It consists of hidden states, observations, transition probabilities, emission probabilities, and initial state distribution. HMM is based on the Markov property and addresses three main problems: evaluation, decoding, and learning. It is widely used in speech recognition, natural language processing, bioinformatics, and robotics. Understanding HMM is essential for modeling sequential and time-dependent data in AI.