IARE

INSTITUTE OF AERONAUTICAL ENGINEERING

Course Title - Logic Programming for Artificial Intelligence

Topic Title – Uncertain Knowledge and Reasoning - Symbolic Reasoning Under Uncertainty: Introduction to Non monotonic Reasoning

Presenter’s Name – Ms. Bidyutlata Sahoo
Presenter’s ID – IARE11028
Department Name – CSE (AI & ML)
Lecture Number - 01
Presentation Date –

Course Outcome

At the end of the course, students should be able to:

CO6: Examine the uncertainty in designing AI systems and propose methods for reasoning.

Topic Learning Outcome

1. ***Understand*** the concept of uncertain knowledge and reasoning to make informed decisions and predictions when all the data is not available.
2. ***Understand*** the nature of uncertainty in AI to model uncertainty in AI systems.
3. ***Use*** non-monotonic reasoning in AI to handle real-world complexity, model human-like reasoning, and make decisions that are adaptive and context-sensitive.

Uncertain Knowledge And Reasoning

Uncertain Knowledge And Reasoning

- *Uncertainty* is a *common challenge in AI*, *particularly when system operate in complex, real-world environments* where complete and accurate information may not always be available.
- *Uncertain knowledge and reasoning* aim to enable AI systems *to make decisions, predictions, and inferences even when faced with ambiguity, incomplete data, or probabilistic events.*
- This is crucial for developing robust and adaptive AI applications.

Why Uncertainty Arises in AI Systems?

- **Incomplete Information**: The system may not have access to all the data needed to make a decision.
- **Noisy Data**: Sensors and data sources can be unreliable, introducing noise into the system.
- **Complex Environments**: Real-world environments are dynamic and unpredictable.
- **Ambiguity**: Multiple interpretations or outcomes might be possible for a given set of inputs.

Handling Uncertainty in AI

- There are several *approaches to represent and reason under uncertainty:*

  a) **Probability Theory (**Bayesian Networks, Hidden Markov Models (HMMs)**)**

  b) **Fuzzy Logic**

  c) **Dempster-Shafer Theory (Belief Functions)**

  d) **Possibility Theory**

Symbolic Reasoning Under Uncertainty

Symbolic Reasoning

- *Reasoning* is *an act of deriving a conclusion from certain premises using a given methodology.*
- *Reasoning* is a *process of thinking; logically arguing; drawing inference.*
- When a system is required to do something, that it has not been explicitly told how to do, it must reason. It must figure out what it needs to know from what it already knows.
- **Example**: Fact-1 : Robins are birds,

  Fact-2 : All birds have wings.

  Then we can ask: DO ROBINS HAVE WINGS?

Human reasoning capabilities

- Broadly it is being **divided into three areas:**
  - *Mathematical Reasoning-* axioms, definitions, theorems, proofs.
  - *Logical Reasoning-* deductive, inductive, abductive, analogical.
  - *Non-logical Reasoning-* linguistic, language.
- Above three reasoning mentioned are very common in every human being, but the ability level depends on education, genetics and environment.
- Intelligent Quotient (IQ)= mathematical + logical reasoning, whereas, Emotional Quotient (EQ) mostly depends on non-logical reasoning capabilities.
- **Logical Reasoning is our major concern in AI.**

Logical Reasoning

- Logic is a language of reasoning. It is a *collection of rules called logic arguments*; we use when doing logical reasoning.
- Logical Reasoning is a *process of drawing conclusions from premises using rule of inference.*

Logical Reasoning

The **study of logic is divided into two**: formal and informal logic.

- The *formal logic* is the study of inference with purely formal content, i.e. where content is made explicit. Eg: Propositional logic and Predicate Logic.
- The *informal logic* is *study of natural language arguments*. The analysis of argument structures in ordinary language is a part of informal logic. The focus lies in distinguishing good arguments (valid) and bad arguments (invalid).

Uncertainty in Reasoning

- The *world is an uncertain place*; *often the knowledge is imperfect which causes uncertainty.* Therefore, reasoning must be able to operate under uncertainty.
- ***AI systems must have ability to reason under conditions of uncertainty.***
  - ✓ Incompleteness knowledge → compensate for lack of knowledge.
  - ✓ Inconsistencies knowledge → Resolve ambiguities and contradictions.
  - ✓ Changing knowledge → update the knowledgebase over time.

Approaches of reasoning under uncertainty

1. **Symbolic reasoning** (Monotonic and Non-monotonic reasoning)
2. **Statistical reasoning** (Probabilistic reasoning / Bayesian Probability theory)
3. Fuzzy logic reasoning

Introduction to Non monotonic Reasoning

Monotonic Reasoning

- *Once the conclusion is taken, then it will remain same* even if we add some other information to existing information in our knowledge base.
- New knowledge from real world can’t be added.
- Decisions are not affected by new facts.
- *Not suitable for real world problem.*
- All old proofs are valid.
- *Used in conventional reasoning systems*.
- **Ex**: Theorem proving.

Monotonic Reasoning

- **Example:**

  Earth revolves around sun. [which is valid can’t be changed even if any new information added later]

  Earth is not round. [new information added later]

Nonmonotonic Reasoning

- *Conclusion may be invalidated if we add some more information to our knowledge base.*
- Helpful in real world scenario.
- **Example:**

Facts:

- Bird’s can fly.
- Penguins can’t fly.
- Alex is bird.

Conclusion: Alex can fly.

Nonmonotonic Reasoning

- **Example:**

Facts:

- Bird’s can fly.
- Penguins can’t fly.
- Alex is bird.
- Alex is a penguin. [new information added later]

Conclusion: Alex can’t fly. [conclusion has been changed]

- Nonmonotonic reasoning ***can’t be used for theorem proving***.

Nonmonotonic Reasoning

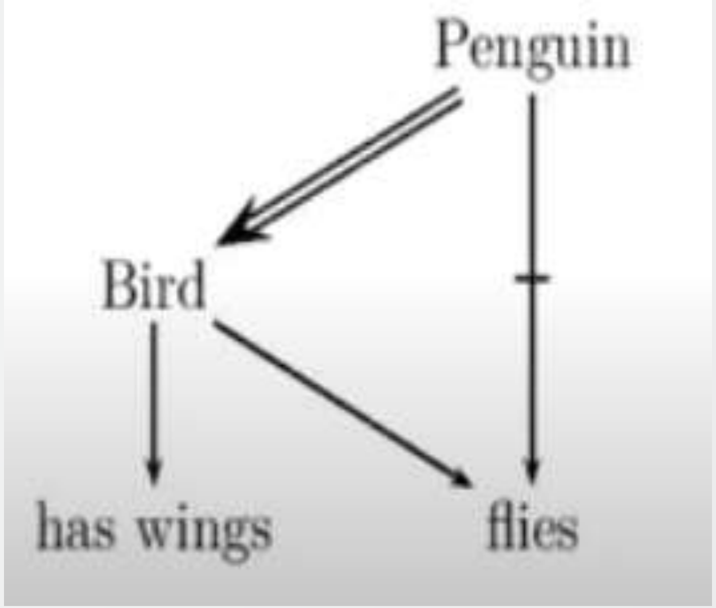
- **Example: (another way of representation)**

Facts:

- Penguin can’t fly.
- Birds has wings.
- Birds can fly.
- Penguin is a bird.

Conclusion:

- Penguin can fly.



Nonmonotonic Reasoning – Key features

- **Default Reasoning**: Assumes defaults (*typical truths*) in the absence of evidence to the contrary.
- **Revisable Conclusions**: Allows *previously drawn conclusions to be retracted* when contradicted by new information.
- **Real-World Applicability**: *Useful in domains where complete information is not always available* (e.g., AI, expert systems).

References

- Stuart Russell and Peter Norvig, “Artificial Intelligence”, 2nd edition, Pearson Education, 2003.
- Saroj Koushik, “Artificial intelligence”.
- NPTEL

IARE

INSTITUTE OF AERONAUTICAL ENGINEERING

Thank You

IARE

INSTITUTE OF AERONAUTICAL ENGINEERING

Course Title - Logic Programming for Artificial Intelligence

Topic Title – Logics for Non monotonic Reasoning, Implementation Issues

Presenter's Name – Ms. Bidyutlata Sahoo
Presenter's ID – IARE11028
Department Name – CSE (AI & ML)
Lecture Number - 02
Presentation Date –

Course Outcome

At the end of the course, students should be able to:

CO6: Examine the uncertainty in designing AI systems and propose methods for reasoning.

Topic Learning Outcome

1. ***Apply*** logics for non-monotonic reasoning which provide formal frameworks to model and manage reasoning processes that are flexible and adaptable to new information.
2. ***Choose*** the appropriate model for representing uncertainty to manage and update the models efficiently.

Logics for Non monotonic Reasoning

Logics for Non monotonic Reasoning

- Basically, Nonmonotonic reasoning is *used for Default reasoning.*
- Default reasoning refers to the general **process of reasoning** where *conclusions are drawn based on typical assumptions or defaults, even when there is incomplete information*.
- There are *two approaches for default reasoning:*

1. Nonmonotonic Logic
2. Default Logic

Logics for Non monotonic Reasoning

Default reasoning - Example

- Question: How many wheels of Johan's car have?
- Answer: 4

- The conclusion is withdrawn if one supplied with the information that one wheel of John's car has just been stolen

Non monotonicity

Logics for Non monotonic Reasoning

1. Nonmonotonic Logic

- Truth of proposition may change when new information's are added and logic may be built to allow the statement to be retracted.
- *Use **modal operator** (**M**) **{read as** " is consistent "}*
- *M is used to allow consistency.*
- **Example:** if we represent "**Normally, birds fly**"

  $\forall x$: Birds(x) → Flies(x))

[Note: This reads: "*If x is a bird, then x flies*". This expresses that the possibility of flying (for birds) can tentatively lead to the conclusion that a bird flies, unless contradicted (e.g., "penguins don't fly").]

Logics for Non monotonic Reasoning

1. Nonmonotonic Logic

- **Example:** Consider the **default rule**:

  "**Typically, birds can fly**."

This is formalized using modal operators as:

$\forall x: Bird(x) \rightarrow M\ Flies(x))$

[Note: This reads: "*If x is a bird, then it is consistent to believe that x flies*". If we later learn that *x is a penguin*, the belief M Flies(x) is invalidated, demonstrating nonmonotonic behaviour. Hence, the belief Flies(x) is **withdrawn** in light of this new evidence.]

Logics for Non monotonic Reasoning

2. Default logic

- **Default logic** can express facts like " *by default, something is true*"; by contrast, **standard logic** can only express that *something is true or that something is false.* This is a problem because reasoning often involves facts that are true in the majority of cases but not always.

- **Example**: " birds typically fly". This rule **can be expressed in standard logic either by** " all birds fly" which is inconsistent with the fact that penguins don't fly or " all birds that are not penguins and not ostriches fly." which requires all exceptions to the rule to be satisfied.

Logics for Non monotonic Reasoning

2. Default logic

- Proposed by **Raymond Reiter**, default logic *allows for reasoning with default assumptions.* Defaults are rules that apply unless contradicted by evidence.
- **Representation**: A default rule has the form:

$$\frac{\text{Prerequisite} : \text{Justification}}{\text{Conclusion}}$$

(The **Prerequisite** must hold, and the **Justification** must be consistent with known facts, for the **Conclusion** to be inferred.)

Logics for Non monotonic Reasoning

2. Default logic

- **Example:** The default rule "birds typically fly" is formalized by the following default:

$$D = \left\{ \frac{\mathrm{Bird}(X) : \mathrm{Flies}(X)}{\mathrm{Flies}(X)} \right\}$$

- i.e., " If X is a bird, and it can be assumed that it flies, then we can conclude that it flies."

Implementation Issues in nonmonotonic reasoning

1. Computational Complexity Issues

Issues

- *Nonmonotonic reasoning often needs to check many possible "scenarios" or combinations of assumptions to decide if something is true.*
- This is like trying to solve a puzzle by testing all possible ways to fit the pieces together, which can take a lot of time and effort if there are too many pieces.

1. Computational Complexity Issues

Example:

- Imagine you have a rule:
    - "Birds usually fly unless there's a reason they can't (like being a penguin)."
- Now, if you have a list of many birds, and for each one you need to figure out:
    1. Are they a typical bird?
    2. Are they a penguin or some other exception?
- If the list is huge, checking all these possibilities becomes extremely slow.

1. Computational Complexity Issues

Solution:

To make this faster:

1. **Approximation:** *Instead of checking all scenarios, focus on the most likely ones* (e.g., assume most birds fly unless there's clear evidence otherwise).
2. **Subsets:** *Break the problem into smaller, manageable parts*. For example, handle only one group of birds at a time instead of checking all at once.

This way, the system works faster without getting overwhelmed by too many possibilities.

2. Handling Conflicting Defaults

Issues:

Sometimes, rules can clash when they apply to the same situation but suggest different answers. This makes it hard for the system to decide what to believe.

2. Handling Conflicting Defaults

Example:

1. **Rule 1:** "Birds can fly." (Default rule for most birds.)
2. **Rule 2:** "Penguins cannot fly." (Exception to Rule 1.)

Now, let's say we learn about Tweety:

- Tweety is a penguin. (So, Tweety shouldn’t fly according to Rule 2.)
- But Tweety is a mutant penguin that might be able to fly. (This goes back to Rule 1.)

The system is now confused:

- Does Tweety fly or not?

2. Handling Conflicting Defaults

Solution:

To resolve this, the system can use **priorities or weights**:

- *Decide which rule is stronger or more specific.*
- For example, Rule 2 ("Penguins cannot fly") might be more specific than Rule 1 ("Birds can fly"), so the system gives it higher priority.
- But if we explicitly state that Tweety is a mutant penguin capable of fly, the *system can prioritize that specific information over the general rule.*

This way, the system can logically decide what to believe based on which rule is more relevant or reliable in the situation.

3. Inconsistencies in Knowledge Base

Issues:

Adding new information can create inconsistencies, requiring the system to retract some conclusions.

3. Inconsistencies in Knowledge Base

Example:

Medical Diagnosis

1. **Initial Knowledge:** "If a patient has a fever and sore throat, they likely have a viral infection."
2. **New Observation:** The patient has a bacterial infection (confirmed by lab results).
3. **Conflict:** The system had concluded it was a viral infection but now learns it's bacterial.
4. **Resolution:** The system revises its diagnosis to: "The patient has a bacterial infection, not a viral one."

3. Inconsistencies in Knowledge Base

Solution:

Use ***belief revision techniques*** to update the knowledge base while maintaining consistency.

4. Scalability

Issues:

Systems with large knowledge bases or many interdependent rules struggle with performance.

Example:

In an expert medical system, considering all possible interactions between symptoms, diagnoses, and treatments can overwhelm computational resources.

Solution:

Use modular reasoning systems or distributed architectures to manage complexity.

5. Dependency Management

Issues:

Nonmonotonic reasoning requires tracking how conclusions depend on rules and assumptions.

Example:

If a conclusion (e.g., "The light is on") depends on multiple conditions ("The switch is on" and "The bulb works"), any change in these conditions requires updating the conclusion.

Solution:

Use dependency graphs to efficiently manage updates and relationships.

References

- Stuart Russell and Peter Norvig, “Artificial Intelligence”, 2nd edition, Pearson Education, 2003.
- Saroj Koushik, “Artificial intelligence”.
- NPTEL

IARE
INSTITUTE OF
AERONAUTICAL ENGINEERING

Thank You

IARE INSTITUTE OF AERONAUTICAL ENGINEERING

Course Title - Logic Programming for Artificial Intelligence

Topic Title – Augmenting a Problem-solver

Presenter's Name – Ms. Bidyutlata Sahoo
Presenter's ID – IARE11028
Department Name – CSE (AI & ML)
Lecture Number - 03
Presentation Date – 06/12/2024

Course Outcome

At the end of the course, students should be able to:

CO6: Examine the uncertainty in designing AI systems and propose methods for reasoning.

Topic Learning Outcome

Use augmenting a problem-solver in AI to enhance the capabilities of existing problem-solving systems by integrating advanced AI techniques.

Augmenting

Problem-solver

Augmenting a Problem-solver

- Augmenting a problem-solver in AI *involves enhancing its capabilities to solve problems more efficiently, effectively, or in a broader range of scenarios.*
- The **goal** is *to improve the system's reasoning, adaptability, and ability to handle complex tasks.*

key aspects of augmentation

1. **Adding Domain Knowledge**

- i.e., *Integrating specific knowledge about the problem domain into the problem-solving system.*
- **For example**: In medical diagnosis, augment the solver with knowledge about diseases, symptoms, and treatments.
- Thereby *reduces the search space* by guiding the system toward relevant solutions.
- And *improves accuracy* by incorporating expert insights.

key aspects of augmentation

2. Using Advanced Search Techniques

- i.e., *Employing more efficient search strategies to explore possible solutions.*
- **For example**: Instead of a simple breadth-first search, *use heuristic searches like A* to prioritize promising paths. Apply iterative deepening or bidirectional search for better performance* in large search spaces.
- Thereby *Speeds up problem-solving.*
- And *reduces computational costs*.

key aspects of augmentation

3. <u>Employing Machine Learning</u>

- i.e., *Incorporating machine learning models to learn from data and improve problem-solving over time.*
- **For example**: Train a model to recognize patterns in previous solutions and suggest shortcuts.
- Thereby *Improves efficiency and accuracy* by leveraging past experiences.
- And *enables the system to adapt to new problems*.

key aspects of augmentation

4. <u>Enhancing Representation</u>

- i.e., *Improving how the problem and its components are represented within the system.*
- **For example**: Use **constraint satisfaction problems (CSPs)** to represent problems with well-defined rules.
- *Makes problem-solving more systematic.*
- *Simplifies complex problems by breaking them into manageable components*.

key aspects of augmentation

5. Incorporating Planning Capabilities

- i.e., *Adding the ability to generate and evaluate plans to solve a problem.*
- **For example**: Use **planning algorithms** to determine the steps needed to achieve a goal, such as creating a travel itinerary.
- *Solves multi-step problems systematically.*
- *Adapts solutions based on changing goals or constraints*.

key aspects of augmentation

6. Dealing with Uncertainty

- i.e., *Enabling the system to handle incomplete or uncertain information.*
- **For example**: Augment with **probabilistic reasoning** (e.g., Bayesian networks) to make decisions based on likelihoods.
- *Makes the system robust to uncertainty*.
- *Adapts solutions based on changing goals or constraints*.

key aspects of augmentation

7. Improving Interaction with Humans

- i.e., *Enabling better collaboration between the AI and human users.*
- **For example**: Add **natural language processing (NLP)** to understand user inputs more effectively.
- *Enhances usability and trust.*
- *Adapts solutions based on changing goals or constraints.*

key aspects of augmentation

8. Parallel and Distributed Computing

- i.e., *Using multiple processors or machines to solve problems faster.*
- **For example**: Divide a large search space into smaller parts and explore them simultaneously.
- *Improves efficiency in solving complex or large-scale problems.*
- *Enables real-time problem-solving in dynamic environments*.

Example - Augmenting a Maze Solver

Basic Solver: A simple depth-first search algorithm finds a path through a maze.

Augmentation:

- ✓Integrate **knowledge** about common maze patterns (e.g., dead ends).
- ✓Add **heuristics** (e.g., prioritize paths closer to the goal).
- ✓Use **parallel processing** to explore multiple paths simultaneously.

Outcome: The augmented solver finds solutions faster and handles more complex mazes than the basic version.

Note:

Augmenting a problem-solver *ensures it is more adaptable, efficient, and capable of handling real-world complexities*, making it a critical step in advancing AI systems.

References

- Stuart Russell and Peter Norvig, “Artificial Intelligence”, 2nd edition, Pearson Education, 2003.
- Saroj Koushik, “Artificial intelligence”.
- NPTEL

Thank You

IARE
INSTITUTE OF
AERONAUTICAL ENGINEERING

Course Title - Logic Programming for Artificial Intelligence

Topic Title – Uncertainty: Review of Probability, Probabilistic Reasoning

Presenter's Name – Ms. Bidyutlata Sahoo
Presenter's ID – IARE11028
Department Name – CSE (AI & ML)
Lecture Number - 04
Presentation Date –

Course Outcome

At the end of the course, students should be able to:

CO6: Examine the uncertainty in designing AI systems and propose methods for reasoning.

Topic Learning Outcome

1. ***Apply*** the concept of probability to develop and deploy AI systems that effectively handle uncertainty and make reliable predictions in complex environments.

2. ***Understand*** probabilistic reasoning in ai to model uncertainty, make informed decisions, and infer hidden information.

All birds Fly !!!

Most of the birds Fly !!!

95% of the birds Fly !!!

Uncertainty

Uncertainty

- Lack of exact information
- Doubtful information

Uncertainty

- Till now, we have learned *knowledge representation using first-order logic and propositional logic **with certainty***, which means we were sure about the predicates.
- With this knowledge representation, we might write **A→B**, which 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.*

Uncertainty

- Artificial intelligence (AI) *uncertainty is when there's not enough information or ambiguity* in data or decision-making.
- AI deals with uncertainty by using models and methods that assign probabilities to different outcomes. Managing uncertainty is important for AI applications like self-driving cars and medical diagnosis, where safety and accuracy are key.

Sources of Uncertainty

- Uncertain Inputs
- Uncertain Knowledge
- Uncertain Outputs

Causes of Uncertainty

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

Methods for Managing Uncertain Information

- Probability
- Bayesian belief network
- Temporal Models
- Hidden Markov Models

Review of Probability

Terminology related to Probability

- **Experiment**
  - Any process that generates outcomes.
  - Example: Tossing a coin, rolling a die.
- **Sample Space (S)**
  - The set of all possible outcomes of an experiment.
  - Example: For a coin toss, S={Heads, Tails}
- **Event**
  - A subset of the sample space.
  - Example: Rolling an even number with a die, A={2,4,6}

Terminology related to Probability (Contd..)

- **Probability (P)**
  - A measure of the likelihood of an event occurring.
  - For event A, $P(A) \in [0,1]$ with $P(S)=1$.
- **Classical Probability**
  - Based on equally likely outcomes.

$$P(A) = \frac{\text{Number of favorable outcomes}}{\text{Total number of outcomes}}$$

Terminology related to Probability (Contd..)

- **Addition Rule**
  - For mutually exclusive events A and B:
  - $P(A \cup B) = P(A) + P(B)$
- **Multiplication Rule**
  - For independent events A and B:
  - $P(A \cap B) = P(A) \cdot P(B)$

Terminology related to Probability (Contd..)

- **Conditional Probability**
  - Probability of A given B:

$$P(A|B) = \frac{P(A \cap B)}{P(B)}, \text{ if } P(B) > 0$$

- **Bayes' Theorem:**
  - Used to reverse conditional probabilities.

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

Probabilistic Reasoning

Probabilistic Reasoning

- Probabilistic reasoning *enables AI systems to make decisions and predictions under uncertainty*.
- It *combines principles of probability theory with logical reasoning*, providing a robust framework for handling incomplete, ambiguous, or noisy information.
- In the real world, there are lots of scenarios, where the certainty of something is not confirmed, such as "It will rain today," "behaviour of someone for some situations," "A match between two teams or two players." These are probable sentences for which we can assume that it will happen but not sure about it, so here we use probabilistic reasoning.

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

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. $0 \leq P(A) \leq 1$, 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.

Probability (Contd..)

We can find the probability of an uncertain event by using the below formula.

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

P(¬A) = probability of a not happening event.
P(¬A) + P(A) = 1.

Probability (Contd..)

Event: Each possible outcome of a variable is called an event.

Sample space: The collection of all possible events is called sample space.

Random variables: Random variables are used to represent the events and objects in the real world.

Prior probability: The prior probability of an event is probability computed before observing new information.

Posterior Probability: The probability that is calculated after all evidence or information has taken into account. It is a combination of prior probability and new information.

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:

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

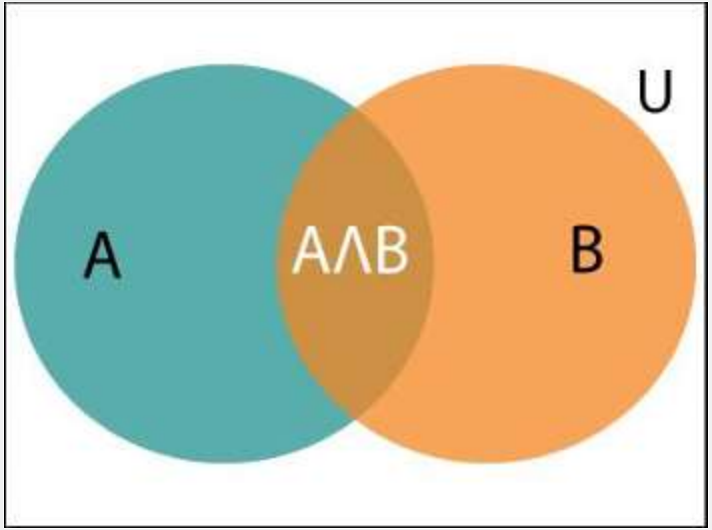
[Where P(A∧B)=Joint probability of A and B

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:

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

Venn Diagram



Venn Diagram - 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.

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

Hence, 57% are the students who like English also like Mathematics.

References

- Stuart Russell and Peter Norvig, “Artificial Intelligence”, 2nd edition, Pearson Education, 2003.
- Saroj Koushik, “Artificial intelligence”.
- NPTEL

INSTITUTE OF AERONAUTICAL ENGINEERING

Thank You

IARE
INSTITUTE OF
AERONAUTICAL ENGINEERING

Course Title - Logic Programming for Artificial Intelligence

Topic Title – Bayesian Networks

Presenter’s Name – Ms. Bidyutlata Sahoo
Presenter’s ID – IARE11028
Department Name – CSE (AI & ML)
Lecture Number - 05
Presentation Date –

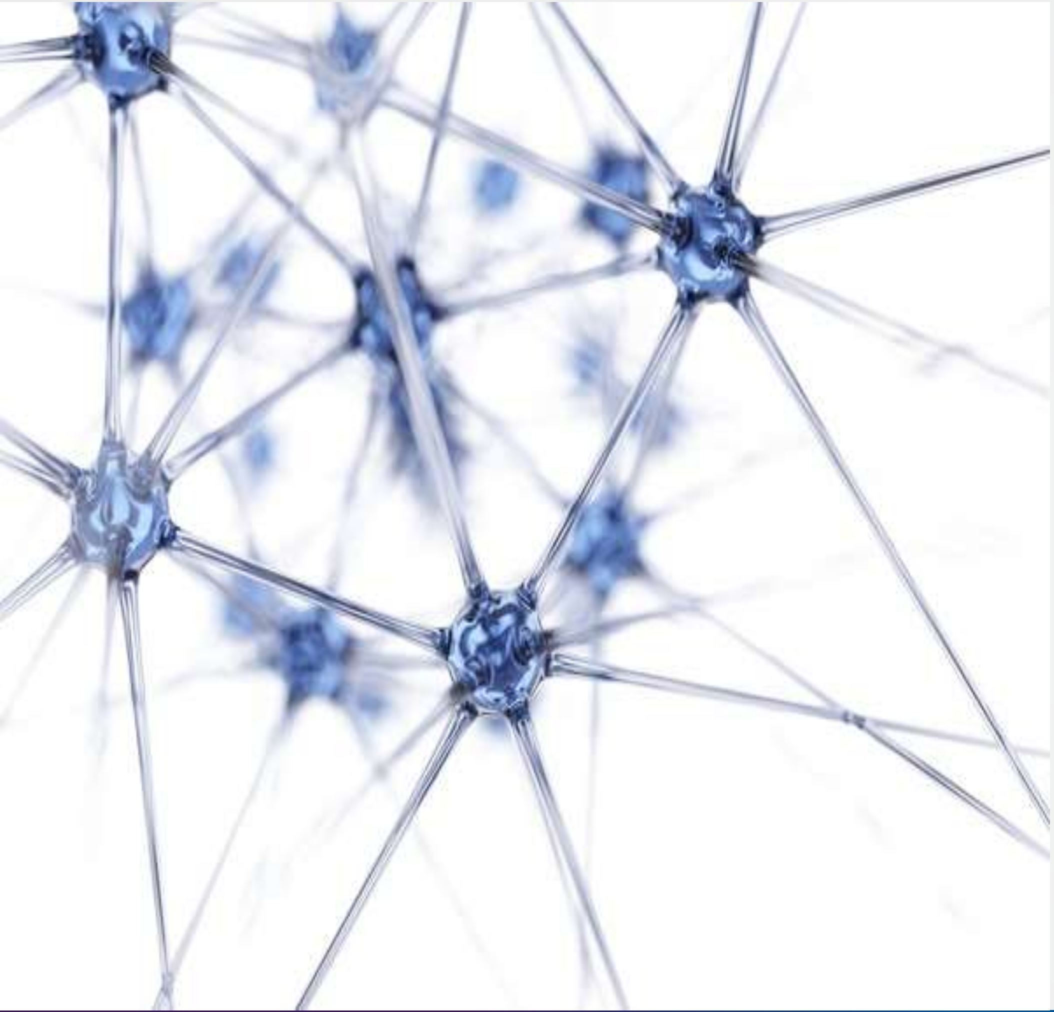
Course Outcome

At the end of the course, students should be able to:

CO6: Examine the uncertainty in designing AI systems and propose methods for reasoning.

Topic Learning Outcome

Understand the graphical structure of Bayesian networks for probabilistic reasoning under uncertainty.



Bayesian Networks

Bayes Theorem

- Bayes theorem is a mathematical concept that *describes the relationship between conditional probabilities of two events*.

Conditional probability: Bayes' Theorem

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

- It is used in Artificial Intelligence and Machine Learning *to build probabilistic models and make decisions based on uncertain data.*

Bayes Theorem

LIKELIHOOD
The probability of "B" being True, given "A" is True

PRIOR
The probability "A" being True. This is the knowledge.

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

POSTERIOR
The probability of "A" being True, given "B" is True

MARGINALIZATION
The probability "B" being True.

Bayes Theorem - Example

Sample Space for events A and B

A holds	T	T	F	F	T	F	T
B holds	T	F	T	F	T	F	F

$P(A) = 4/7$ $P(B) = 3/7$ $P(B|A) = 2/4$ $P(A|B) = 2/3$

Is Bayes Theorem correct?

$P(B|A) = P(A|B)P(B) / P(A) = (2/3 * 3/7) / 4/7 = 2/4$ ➔ CORRECT

$P(A|B) = P(B|A)P(A) / P(B) = (2/4 * 4/7) / 3/7 = 2/3$ ➔ CORRECT

Bayesian Networks / Bayesian Belief Networks

- Bayesian belief network is the key computer *technology for dealing with probabilistic events and to solve a problem which has uncertainty.*
- Bayesian belief network is a useful way *to represent probabilistic models and visualize them.*
- Here *Probabilistic models determine the relationship between variables*, and then you can calculate the various probabilities of those two values.

Bayesian Networks - Definition

- "*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 - Applications

- 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 Networks - Structure

- Bayesian Network consists of *two parts:*
  - **Directed Acyclic Graph (DAG)**
  - **Conditional Probability Table (CPT)**

Bayesian Networks - Structure

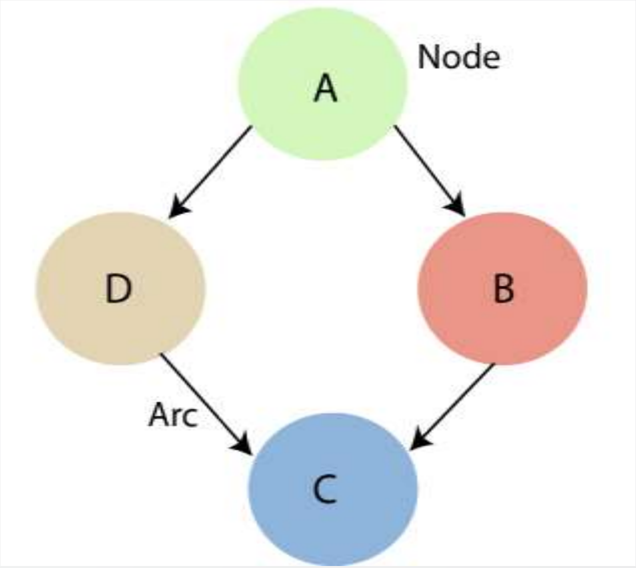
A Bayesian network graph is made up of nodes and Arcs (directed links), where:

- *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.*

Bayesian Networks - Graph

- In the given diagram, *A, B, C, and D are random variables / Hypothesis* 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 Networks - Example

Here there are **3 events-** Rain, Dogbarks & Cathides.

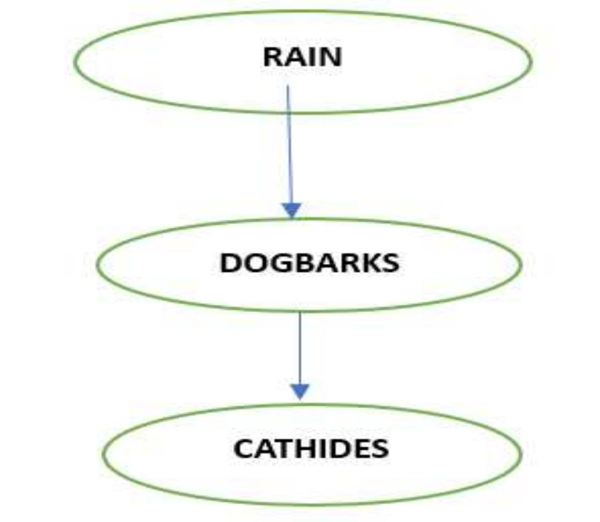
Rain is the parent of Dogbarks & Dogbarks is the parent of Cathides.

i.e., ***when there is rain, dog barks and when dog barks, cat hides.***

So here we can calculate the probability of Dogbarks w.r.t Rain and probability of Cathides w.r.t Dogbarks.

Here *each event* is *called as random variable/ Hypothesis.*

[Note: probability should be calculated for child node w.r.t parent node]



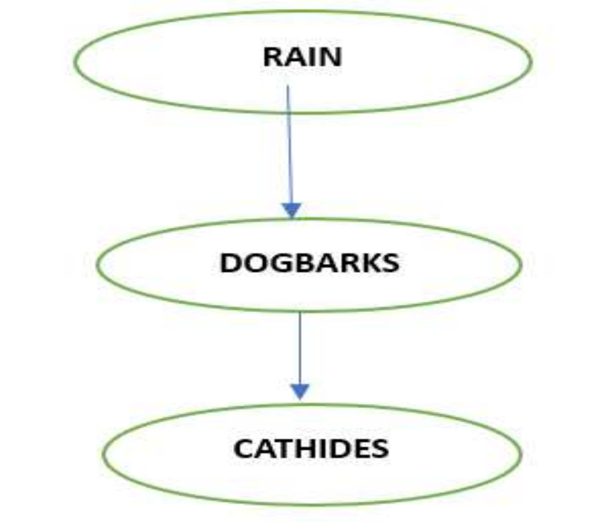
Bayesian Networks - Example

- **Directed Acyclic Graph (DAG)**

  In this example all the 3 events are connected through directed edges without forming any cycle.

- **Conditional Probability Table (CPT)**

	R	~R
B	9/48	18/48
~B	3/48	18/48



That means

(B=T & R=T) = 9/48 = 0.19

(B=T & R=F) = 18/48 = 0.375

(B=F & R=T) = 3/48 = 0.06

(B=F & R=F) = 18/48 = 0.375

Bayesian Networks - Components

- Causal Component
- Actual numbers

Each node in the Bayesian network has conditional probability distribution P(Xi |Parent(Xi)) , *which determines the effect of the parent on that node.*

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

Joint Probability Distribution

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.*

P[x1, x2, x3,....., xn],

it can be written as the following way in terms of the joint probability distribution.

= P[x1| x2, x3,....., xn]P[x2, x3,....., xn]

= P[x1| x2, x3,....., xn]P[x2|x3,....., xn]....P[xn-1|xn]P[xn].

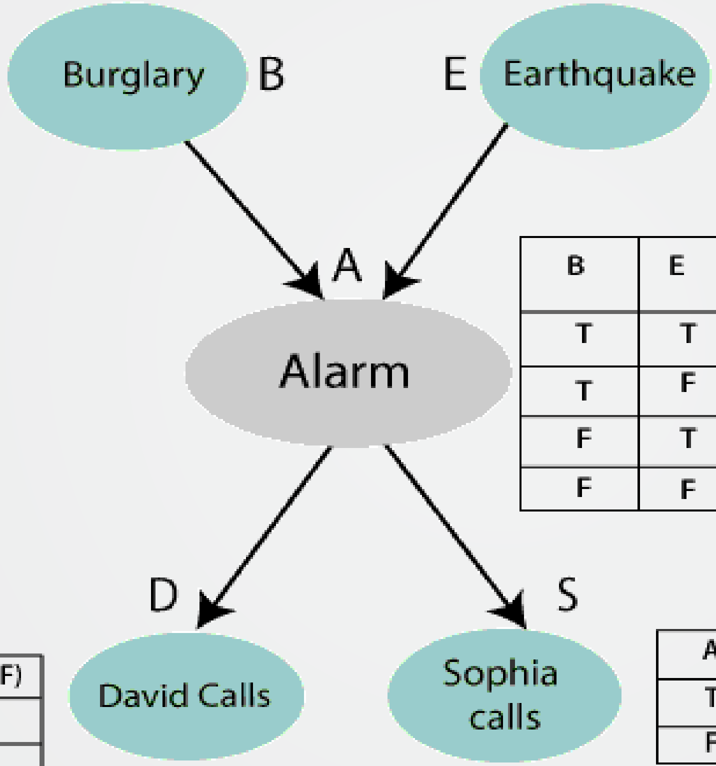
In general for each variable X_i, we can write the equation as:

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

Bayesian Networks – Numerical Problem

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 neighbours 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.

Bayesian Networks – Numerical Problem (Contd..)



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

Bayesian Networks – Numerical Problem (Contd..)

List of all events occurring in this network:

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

Bayesian Networks – Numerical Problem (Contd..)

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.

Bayesian Networks – Numerical Problem (Contd..)

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

Bayesian Networks – Numerical Problem (Contd..)

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

Bayesian Networks – Numerical Problem (Contd..)

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

Bayesian Networks – Numerical Problem (Contd..)

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.

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

$P(S, D, A, \neg B, \neg E) = P(S|A) * P(D|A) * P(A|\neg B \wedge \neg E) * P(\neg B) * P(\neg E)$

$= 0.75 * 0.91 * 0.001 * 0.998 * 0.999$

$= 0.00068045.$

References

- Stuart Russell and Peter Norvig, “Artificial Intelligence”, 2nd edition, Pearson Education, 2003.
- Saroj Koushik, “Artificial intelligence”.
- NPTEL

INSTITUTE OF
AERONAUTICAL ENGINEERING

Thank You

Course Title - Logic Programming for Artificial Intelligence

Topic Title – Temporal Models

Presenter's Name – Ms. Bidyutlata Sahoo
Presenter's ID – IARE11028
Department Name – CSE (AI & ML)
Lecture Number - 06
Presentation Date –

Course Outcome

At the end of the course, students should be able to:

CO6: **Examine** the uncertainty in designing AI systems and propose methods for reasoning.

Topic Learning Outcome

Understand temporal models to analyze time-series data.

Temporal Models

Temporal Models

- **Temporal models** are tools *used to represent, analyze, and predict data that change over time.*
- They are *essential for* systems *where the sequence and time of events matter*, such as *speech, video, financial data, or sensor readings.*
- Temporal models capture **temporal dependencies**, meaning that *the current state of the system is dependent on its past states.*

Temporal Models

- Agents in uncertain environments must be able to keep track of the current state of the environment.
- This is difficult in partial and noisy perceptions and uncertainty.
- Because environment changes over time.
- Agent will be able to obtain only a probabilistic assessment of the current situation.

Temporal Models

There are two important concepts in temporal models:

- **Time and Uncertainty**
- **Inferences in temporal models**

Temporal Models

Time and Uncertainty:

- A changing world is modelled using a random variable.
- Model each aspect of the world state at each point in time.

Temporal Models

<u>Time and Uncertainty:</u>

Example:

- Evidence: Recent insulin doses, food intake, blood sugar measurements and other physical signs.
- Assess the current state of the patient, including the actual blood sugar level and insulin level.

Temporal Models

Time and Uncertainty:

- The doctor (or patient) makes a decision about the patient's food intake and insulin dose.
- Blood sugar level can change rapidly over time, depending on one's recent food intake and insulin doses, the time of day.
- To assess the current state from the history of evidence and to predict the outcome of treatment actions, we must model these changes.

Temporal Models

- Time and Uncertainty (under this 2 important topics)
  - ✓ State and observations
  - ✓ Stationary processes and Markov assumptions
- Inferences in temporal models

Temporal Models

State and observations

- The process of change *can be viewed as a series of snapshots, describes the states of the world at a particular time.*
- Each snap shot or time slice contains a set of random variables, some of which are observable and some of which are not.

Temporal Models

Stationary processes and Markov assumptions

- *With the set of state and evidence variables for a given problem, the next step is to specify the dependencies among the variables.*
- One obvious choice is *to order the variables in their natural temporal order*, since **cause** usually precedes **effect** and we prefer to add the variables in casual order.

Temporal Models - Types

- Hidden Markov Models (HMM)
- Recurrent Neural Networks (RNN)
- Dynamic Bayesian Networks (DBN)
- Kalman Filter (State-Space Model)

Temporal Models - Applications

- Speech Recognition
- Video and Gesture Recognition
- Stock Market Prediction
- Weather Forecasting
- Healthcare

Temporal Models - Advantages

- Capturing Temporal Dependencies
- Real-Time Predictions
- Probabilistic Framework

Temporal Models - Limitations

- Complexity
- Memory Issues
- Data Sparsity

References

- Stuart Russell and Peter Norvig, “Artificial Intelligence”, 2nd edition, Pearson Education, 2003.
- Saroj Koushik, “Artificial intelligence”.
- NPTEL

IARE
INSTITUTE OF
AERONAUTICAL ENGINEERING

Thank You

IARE
INSTITUTE OF AERONAUTICAL ENGINEERING

Course Title - Logic Programming for Artificial Intelligence

Topic Title – Hidden Markov Models

Presenter’s Name – Ms. Bidyutlata Sahoo
Presenter’s ID – IARE11028
Department Name – CSE (AI & ML)
Lecture Number - 07
Presentation Date –

Course Outcome

At the end of the course, students should be able to:

CO6: Examine the uncertainty in designing AI systems and propose methods for reasoning.

Topic Learning Outcome

Apply Hidden Markov Models to model and analyze sequential and time-series data.

Hidden Markov Models

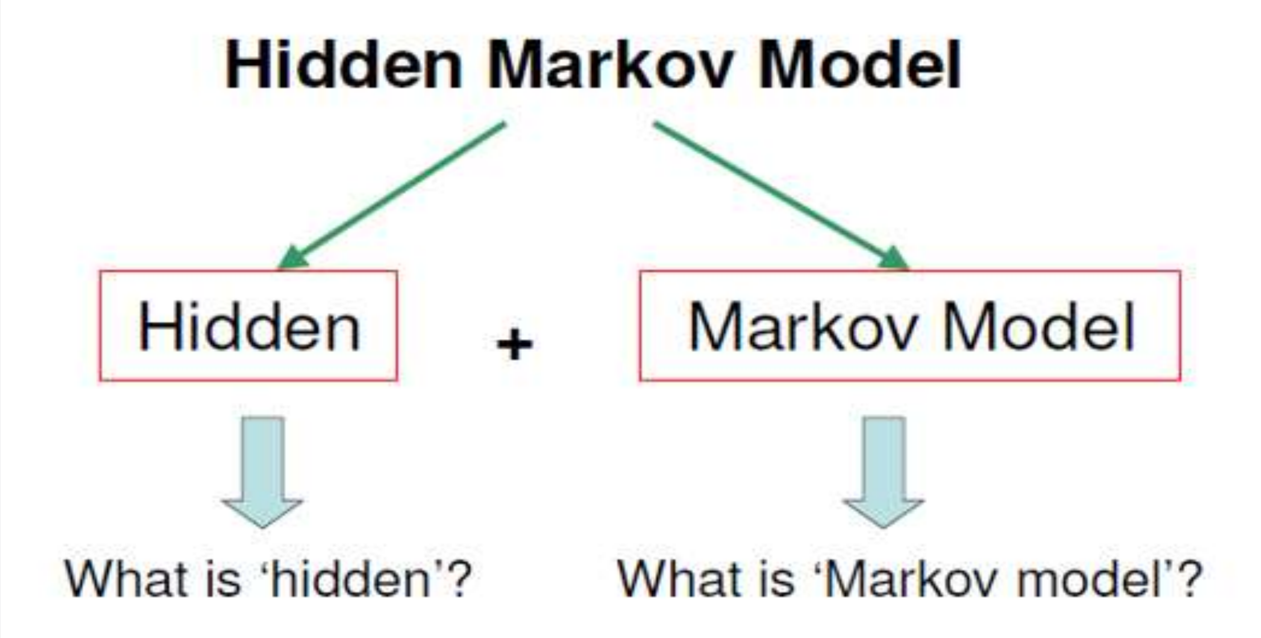
Hidden Markov Models

- A hidden Markov model (HMM) is a *statistical model*, in which the *system being modelled is assumed to be a Markov process* (Memoryless process: its future and past are independent) *with hidden states.*
- A **Hidden Markov Model (HMM)** is a **statistical model** *used to describe systems that have a sequence of observable events generated by a sequence of hidden (unobservable) states.*
- It is widely applied in areas like *speech recognition, natural language processing, and bioinformatics.*

Hidden Markov Models

- Has a set of states each of which has limited number of transitions and emissions.
- Each transition between states has an assigned probability.
- Each model starts from start state and ends in end state.

Hidden Markov Models



Hidden Markov Models

- **Markov models**:
    - Talk about weather.
    - Assume there are three types of weather.

Hidden Markov Models

- **Markov models**:
  - Weather at day **n** is : $q_n \in \{sunny, rainy, foggy\}$

  - q_n depends on the known weathers of the past days (q_{n-1}, q_{n-2},…)

Hidden Markov Models

- **Markov models**:
  - We want to find that : $P(q_n | q_{n-1}, q_{n-2}, \ldots, q_1)$
  - That means *given the past weathers what is the probability of any possible weather of today?*

Hidden Markov Models

- **Markov models**:

  <u>For example:</u>

  - if we knew the weather for last three days was:

  - the probability that tomorrow would be  is:

Hidden Markov Models

- **Markov models**:

  Therefore, make a simplifying assumption Markov assumption:

  - For sequence: $\{q_1, q_2, \ldots, q_n\}$

  $$P(q_n | q_{n-1}, q_{n-2}, \ldots, q_1) = P(q_n | q_{n-1})$$

  - the weather of tomorrow only depends on today (first order Markov model)

Hidden Markov Models

- **Example:**
  - Suppose that you are locked in a room for several days.
  - You try to predict the weather outside.
  - The only piece of evidence you have is whether the person who comes into the room bringing your daily meal is carrying an umbrella or not.

Hidden Markov Models

- **Example (Contd..):**
  - Assume probabilities as seen in the table:

Weather	Probability of umbrella
Sunny	0.1
Rainy	0.8
Foggy	0.3

Probability $P(x_i|q_i)$ of carrying an umbrella ($x_i = \text{true}$) based on the weather q_i on some day i

Hidden Markov Models

- **Example (Contd..):**
  - Finding the probability of a certain weather.

$$q_n \in \{sunny, rainy, foggy\}$$

  - is based on the observations $\mathbf{x_i}$

Hidden Markov Models

- **Example (Contd..):**
  - Using Bayes rule:

$$P(q_i|x_i) = \frac{P(x_i|q_i)P(q_i)}{P(x_i)}$$

  - For n days:

$$P(q_1,\ldots,q_n|x_1,\ldots,x_n) = \frac{P(x_1,\ldots,x_n|q_1,\ldots,q_n)P(q_1,\ldots,q_n)}{P(x_1,\ldots,x_n)}$$

Hidden Markov Models - Applications

- *Speech Recognition* (Map audio signals to spoken words.)
- *Natural Language Processing* (Tasks like part-of-speech tagging or named entity recognition.)
- *Bioinformatics* (DNA sequencing and protein structure prediction.)
- *Gesture Recognition* (Identify human gestures from motion data.)

Hidden Markov Models - Advantages

- *Models sequential and temporal data effectively.*
- *Handles uncertainty probabilistically.*

Hidden Markov Models - Limitations

- *Assumes the Markov property* (future depends only on the current state).
- *Assumes conditional independence of observations given the state*.
- *May struggle with long-term dependencies.*

References

- Stuart Russell and Peter Norvig, “Artificial Intelligence”, 2nd edition, Pearson Education, 2003.
- Saroj Koushik, “Artificial intelligence”.
- NPTEL

IARE
INSTITUTE OF
AERONAUTICAL ENGINEERING

Thank You