Parul® University

Artificial Intelligence

Parul®
University

DIGITAL LEARNING CONTENT

CHAPTER-4

Uncertainty

Non monotonic reasoning

You often make **guesses or assumptions** based on what you know. Then, if new information comes along that proves your guess wrong, you change your mind! That's the essence of **non-monotonic reasoning**.

•Monotonic Reasoning:

"Once True, Always True"

➢In monotonic reasoning, if a conclusion is drawn from a set of facts, adding *more* facts will *never* make that conclusion false. You only ever add to your knowledge; you never have to retract anything.

➢It's like building with solid bricks – once a brick is laid, it's permanent.

Monotonic reasoning:

Example 1: Mathematical Facts

- **Initial Facts:**
  - Fact A: 2+2=4
  - Fact B: 4<5

- **Conclusion:** Based on these facts, we can conclude 2+2<5.

- **Adding New Information:**
  - New Fact C: 5×3=15
  - New Fact D: The capital of France is Paris.

- **Result:** Even with these new facts, our original conclusion (2+2<5) remains absolutely true. It's not affected by the new information. The set of true statements just grows.

Example 2: Basic Deductions

- **Initial Facts:**
  - All humans are mortal.
  - Socrates is a human.

- **Conclusion:** Socrates is mortal.

- **Adding New Information:**
  - The sky is blue.
  - Dogs have four legs.

- **Result:** The conclusion "Socrates is mortal" is still true. The new facts don't change or contradict it in any way.

•Non-Monotonic Reasoning:

"I Might Change My Mind Later“

➢In non-monotonic reasoning, a conclusion drawn from a set of facts *can be withdrawn or changed* if new information is added. This is how we often reason in the real world, making educated guesses based on incomplete information.

➢ It's like drawing on a whiteboard – you can erase and redraw as you get new data

•**Basic Concepts:**

➢**Default Assumptions:** We often assume things are true by default unless we have evidence to the contrary (e.g., "Birds can fly").

➢**Incomplete Information:** We make decisions based on the information we have, even if it's not exhaustive.

➢**Changing Environments:** The world changes, and our conclusions need to adapt.

•**Advantages:**

➢Handle incomplete and uncertain information.

➢Reason with default assumptions.

➢Revise beliefs in the face of new evidence.

➢Model common-sense reasoning.

Example 1: The Bird That Flies (or Doesn't!)

- **Initial Facts:**
  - Most birds can fly.
  - Tweety is a bird.

- **Initial Conclusion (Default Assumption):** Tweety can fly.

- **Adding New Information:**
  - New Fact: Tweety is a penguin.
  - New Rule: Penguins are birds that cannot fly.

- **Result:** Our initial conclusion "Tweety can fly" is now **invalidated**. With the new information, we must retract that conclusion and update our belief to "Tweety cannot fly."

Example 2: The Doctor's Diagnosis

Initial Facts:
Patient has a cough.
Patient has a fever.

Initial Conclusion (Probable Diagnosis): The patient has the flu.

Adding New Information:
New Fact: Blood tests show the patient has a bacterial infection.

Result: The initial conclusion "The patient has the flu" is **withdrawn**. The new, more specific information points to a different diagnosis, likely bacterial pneumonia, and requires a different treatment.

Parul® University

NAAC A++ ACCREDITED UNIVERSITY

Logics for non monotonic reasoning(NMLs)

•Imagine we are building a thinking robot. These logics are like different programming tricks we will teach it so it doesn't get stuck when the world isn't perfectly neat and tidy.

1. Default Logic:

- The "Unless Proven Otherwise" Rule

This is what I assume, **unless someone tells me otherwise**." It's great for common-sense rules that have exceptions.

How it works (Robot's thought process):

"If I see an X, I'll assume Y is true about it."

"But wait, is there *any* information that says Y is *not* true about X? If so, I'll ignore my assumption."

- **Example: The coffee shop**

- **Default Rule:** "If a coffee shop is open, **unless there's a sign saying 'Closed for Cleaning'**, I'll assume I can *buy coffee* there.“

- **Situation 1:** You walk by "The Daily Grind." It's open.
  - **Robot's thought:** "It's open. No 'Closed' sign. I assume I **can buy coffee**." (dlt)efa

- **Situation 2 (New Info):** You walk by "The Daily Grind" again. It's open, but now there's a big sign in the window: "Closed for Cleaning."
  - **Robot's thought:** "It's open. BUT there's a 'Closed' sign! That 'proves otherwise' for my default. So, I **cannot buy coffee**." (Retracts previous conclusion)

- **Example: Traffic Lights**

- **Default Rule:** "When the traffic light is green, **unless a police officer is signaling otherwise**, I will *go*."

- **Situation 1:** Light is green. No officer.
  - **Robot's thought:** "Green light, no officer. I **go**."

- **Situation 2 (New Info):** Light is green. A police officer is waving you to stop.
  - **Robot's thought:** "Green light. BUT there's an officer telling me to stop! That 'proves otherwise.' I **do not go**."

2. Autoepistemic Logic:
The "Based on What I Know (or Don't Know)" Rule

This logic is about what the AI *believes* or *knows*, and specifically, what it *doesn't know*. It's like saying: "If I *haven't learned anything* that says this isn't true, then I'll believe it.“

How it works (Robot's thought process):
"If A is true, AND I **don't know that B is false**, then I'll conclude C." (It's about the *absence* of contradictory knowledge.)

Example: The Package Delivery

Rule: "If I sent a package, AND I **haven't received a 'delivery failed' notification**, then I'll assume the package *was delivered*."

Situation 1: You sent a package. It's been a few days. No notification.
Robot's thought: "Sent package. No 'failed' notice. I assume the package **was delivered**."

Situation 2 (New Info): A week later, you get an email: "Delivery Failed: Recipient not home."
Robot's thought: "Sent package. NOW I *have* received a 'delivery failed' notice. My rule's condition is false. I **retract my conclusion** that it was delivered."

Example: Your Friend's Availability

Rule: "If it's Saturday, AND I **don't know that my friend Alex is busy**, then I'll assume Alex is *available to hang out*."

Situation 1: It's Saturday. You haven't talked to Alex all week.
Robot's thought: "It's Saturday. Do I know Alex is busy? No. So, Alex is **available to hang out**." (Makes assumption based on lack of knowledge)

Situation 2 (New Info): It's Saturday. You get a text from Alex: "So sorry, I'm working all day!"
Robot's thought: "It's Saturday. Do I know Alex is busy? YES! I just got a text. So, my rule's condition ('don't know Alex is busy') is now false. I **retract my conclusion** that Alex is available."

3. Circumscription:
"Minimizing the Weirdness“

It's about assuming the *simplest* possible world, with the *fewest exceptions or abnormalities*.

It's like saying: "Don't assume anything unusual is happening unless you absolutely have to.“

How it works (Robot's thought process):
"I have a set of facts and general rules."
"I will try to interpret these facts in a way that creates the **smallest possible number of 'exceptions' or 'abnormalities'**."
"If a new fact *forces* me to acknowledge an exception, then I will, but only then.”

Example: The Pet Cat

General Rule: "Cats are clean animals, *unless* they have some kind of 'abnormality' (like being sick or very old).“

Situation 1: You have a new cat, "Whiskers."
Robot's thought: "I want to minimize 'abnormalities.' Is there any reason Whiskers is abnormal? No. So, I assume Whiskers is **not abnormal**, and therefore, Whiskers is **clean**." (Simplest world: no abnormal cats unless told)

Situation 2 (New Info): You come home and Whiskers has thrown up on the carpet.
Robot's thought: "Oh! This new fact (vomit) *forces* me to acknowledge that Whiskers *is* 'abnormal' in some way (sick). Since Whiskers is abnormal, my rule 'cats are clean' doesn't apply right now. My previous conclusion ('Whiskers is clean') is **no longer certain**."

key differences between default logic ,autoepistemic logic and circumscription logic

Default Logic: Assumes things are true by default *unless directly contradicted by new facts*.

Autoepistemic Logic: Reasons about what the system *knows* and *doesn't know*, making assumptions based on the *absence* of knowledge.

Circumscription: Assumes that exceptions or abnormal conditions are *minimal* unless explicitly forced by evidence.

Forward rules, and Backward rules

•If we are trying to figure out, if someone has a certain illness or if a machine is broken. In Artificial Intelligence, we often use "rules" (like "IF symptom A and symptom B, THEN illness C") and "facts" (like "The patient has symptom A"). But real life isn't always black and white; there's often *uncertainty*. Maybe you're only 80% sure about a symptom, or a rule isn't always true, only 90% of the time.

•"Forward Rules" and "Backward Rules" are just two different ways an AI brain (or "inference engine") figures things out, especially when dealing with this uncertainty.

Cont..

Think of uncertainty as a "belief level" or a "likelihood." Instead of saying something is "True" or "False," we might say it's "0.8 likely" or "we are 70% sure.“

Why uncertainty?

- **Missing info:** You don't have all the pieces of the puzzle.
- **Fuzzy info:** Some things are vague ("warm" isn't an exact temperature).
- **Probabilities:** Things don't always happen. If it rains, you *might* need an umbrella, but not always.

Parul® University

NAAC A++ ACCREDITED UNIVERSITY

DIGITAL LEARNING CONTENT

Forward Rules

How it works:

1)Start with everything you know (your "facts") and how sure you are about each fact.

Example: "The patient has a fever (I'm 90% sure)." "The patient has a cough (I'm 70% sure)."

2)Look at your rule book. Find rules where the first part (the "IF" part) matches what you know.

Rule: "IF Fever AND Cough, THEN Flu (I'm 80% sure this rule is correct)."

3)Use the rule to figure out new things, and calculate how sure you are about the new thing.

Calculation: If you're 90% sure about fever and 70% sure about cough, you're only as sure as the *least certain* part of the "AND" (so, 70% sure about "Fever AND Cough"). Then, multiply that by how sure the rule itself is (70% * 80% = 56%).

New Fact: "The patient has Flu (I'm 56% sure)."

Cont..

4)Keep going. Now you have "Flu" as a new fact. See if "Flu" helps you trigger any other rules.

5)Combine if needed. If two different rules point to the same conclusion, you combine their certainty levels to get an overall belief. (Like, if another rule also suggests "Flu" with 30% certainty, you'd have a combined higher certainty for "Flu".)

6)Stop when you can't learn anything new.

Analogy: You're a baker with ingredients (facts). You apply recipes (rules) to create new dishes (conclusions). You keep baking until you've made everything you can with your ingredients.

When to use: When you have a lot of initial information and you want to see what all the possible outcomes or diagnoses could be.

Backward Rules

- **How it works:**

1)Start with a specific question or goal.

- *Example:* "Does the patient have 'Severe Flu'?"

2)Look at your rule book. Find rules that *conclude* your goal.

- *Rule:* "IF Flu AND Muscle Aches, THEN Severe Flu (I'm 90% sure this rule is correct)."

3)Now, those "IF" parts become your new mini-goals. To prove "Severe Flu," you first need to prove "Flu" and "Muscle Aches."

Cont..

4)Check each mini-goal.

- *For "Muscle Aches":* Is it a known fact? Yes, "Patient has Muscle Aches (I'm 80% sure)." Great, one mini-goal met.
- *For "Flu":* Is it a known fact? No. So, you go back to step 2 and try to find rules that conclude "Flu."
  - (Recursively, you might find "IF Fever AND Cough, THEN Flu" and "IF Sore Throat, THEN Flu".)
  - You'd then check if "Fever," "Cough," and "Sore Throat" are known facts and how certain they are.

5)As you prove each mini-goal, you pass the certainty level back up the chain.

6)Combine the certainties as you move back up, until you get a final certainty for your main goal.

Cont..

•**Analogy:** You want to bake a cake (your goal). You look at the cake recipe to see what you need (flour, eggs, etc. - your sub-goals). If you don't have flour, you then look up the "make flour" recipe (another sub-goal), and so on, until you get to basic ingredients you already have.

•**When to use:** When you have a specific problem or question you're trying to answer, and you want to efficiently find out what evidence you need.

Parul® University

NAAC A++ ACCREDITED UNIVERSITY

Difference between forward and backward rules

1	Feature	Forward Chaining	Backward Chaining
2	Approach	Data-driven, bottom-up	Goal-driven, top-down
3	Starting Point	Known facts	A specific goal or hypothesis
4	Direction	From facts to conclusions	From conclusions/goals to facts
5	When to Use	When you have initial facts and want to see what conclusions can be drawn.	When you have a specific goal to prove or verify.
6	Efficiency	Can generate many irrelevant conclusions if the knowledge base is large.	More focused, only exploring relevant paths to the goal.
7	Example AI	Expert systems for monitoring, control, analysis.	Expert systems for diagnosis, planning, query answering (e.g., MYCIN, Prolog).

Justification Based Truth Maintenance System(JTMS)

•When your brain learns something new, you might realize an old belief no longer makes sense. A JTMS is like a special part of an AI's brain that helps it:

1.Remember *why* it believes something. (e.g., "I believe the light is on *because* I flipped the switch and the bulb isn't broken.") These "whys" are called **justifications**.
2.Automatically update beliefs when new information arrives. If you learn the bulb *is* broken, the JTMS instantly knows the light is probably *not* on anymore, without having to rethink everything from scratch.

It's all about keeping your AI's "beliefs" consistent and up-to-date, especially when information changes.

Example

Belief 1: "It's cloudy" (Certainty: 0.9)
Belief 2: "It's windy" (Certainty: 0.7)
Rule: "IF cloudy AND windy THEN storm" (Rule strength: 0.8)
JTMS records: "Storm" is believed *because* "cloudy" and "windy" are believed.

Uncertainty calculation:
min(0.9,0.7)=0.7 (from "AND" part)
$0.7 * 0.8 = 0.56$
So, the AI believes "Storm" with 0.56 certainty.

Example

•Now, imagine you get new information: "It's actually *not* very windy, only 0.3 certainty."

JTMS action: It sees "windy" changed. It knows "storm" depends on "windy.“

Uncertainty re-calculation:

min(0.9 (cloudy),0.3 (new windy))=0.3

0.3∗0.8=0.24

The AI now updates "Storm" to 0.24 certainty.

JTMS provides the "map" of how beliefs connect. Adding uncertainty means each belief on that map also carries a "how sure" value, and the JTMS helps propagate and update these "how sure" values whenever the information changes.

Advantages And Disadvantages(JTMS)

Adv:

- Non-Monotonic Reasoning: Reasoning adapts based on new information. Conclusions aren't absolute and can be revised.
- Efficient Belief Revision: The system focuses on revising relevant beliefs based on justification dependencies.
- Reasoning about Assumptions: Analyze hypothetical scenarios by making assumptions and tracking consequences within the JTMS framework.

Disadv:

- Complexity: Managing consistency can become computationally expensive with a large number of beliefs and justifications.
- Limited Explanations: The primary focus is on maintaining consistency, explanations for specific beliefs might not be readily available.

Parul® University

NAAC A++ ACCREDITED UNIVERSITY

DIGITAL LEARNING CONTENT

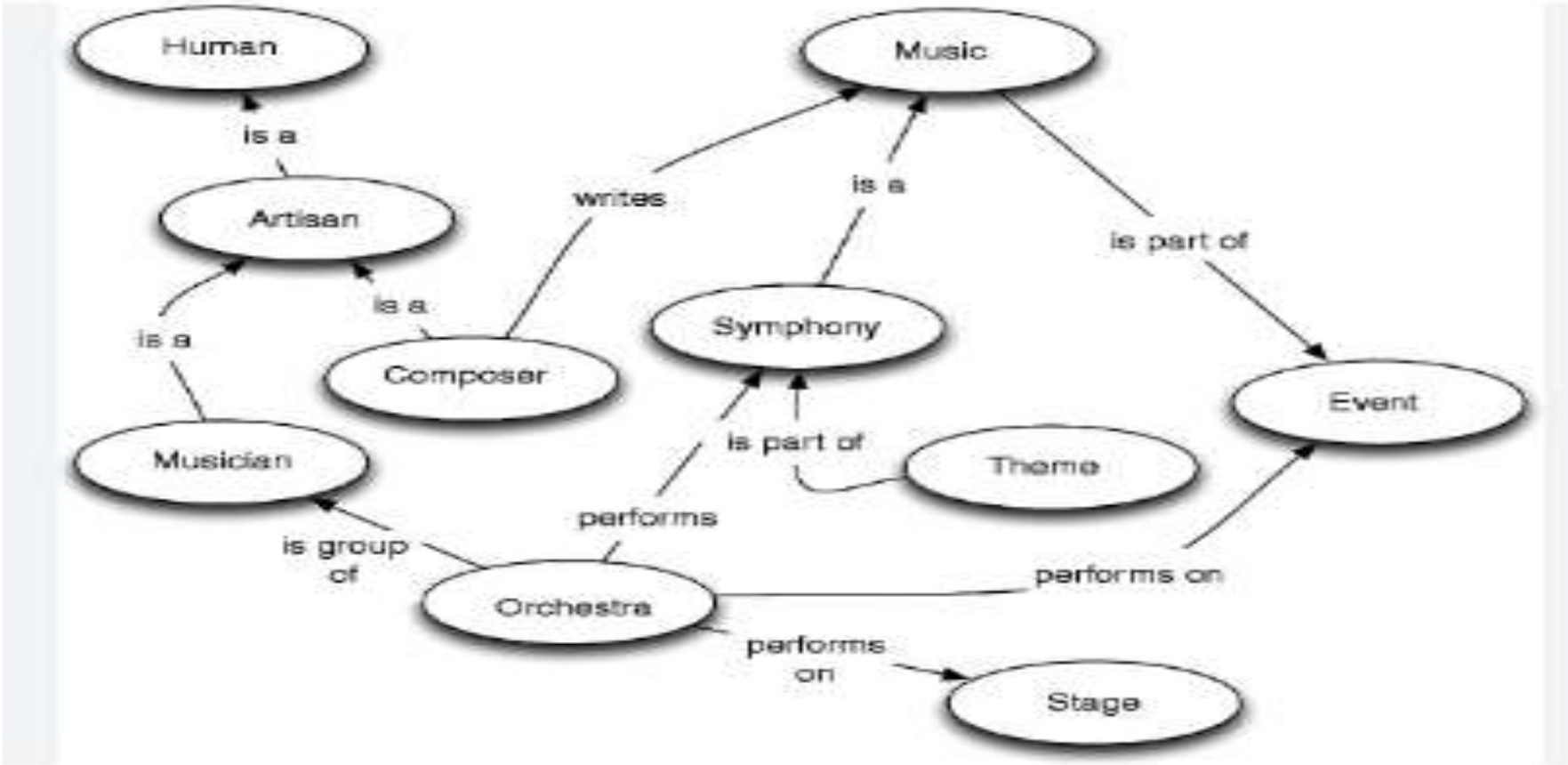
Semantic Nets(Knowledge Map)

•**A giant spider web or a mind map where:**

Nodes (circles): Are like concepts, objects, or ideas. (e.g., "Dog," "Animal," "HasFur," "Barks").

Links/Edges (lines): Show how these concepts are related. (e.g., "is-a," "has-a," "can-do").

•Semantic Net is just a visual way for an AI to store and organize its knowledge, showing how different pieces of information are connected.


Human
is a
Artisan
is a
is a
Musician
is a
Composer
writes
Music
is a
Symphony
is part of
Event
is part of
Theme
performs
is group of
Orchestra
performs on
performs on
Stage

Example

- Node: "Dog"
- Link: "is-a"
- Node: "Mammal"
- Node: "Dog"
- Link: "has-a"
- Node: "Tail"
- Node: "Dog"
- Link: "can-do"
- Node: "Bark"

Parul® University

NAAC A++ ACCREDITED UNIVERSITY

Combining Semantic Nets with Statistical Reasoning in Uncertainty

•When you combine these, you're essentially putting **probabilities or certainty factors on the links or even the nodes** in your semantic net. This allows the AI to:

1)Store uncertain relationships: Instead of just "Dogs bark," it can store "Dogs bark (with 0.9 probability)."
2)Reason with incomplete or fuzzy information: If it only knows "something is a dog," and it knows "dogs bark with 0.9 probability," it can infer that *this something* probably barks, but isn't 100% sure.
3)Calculate the overall certainty of a conclusion: As the AI moves through its knowledge web (the semantic net), it combines the probabilities along the paths to figure out how certain its final answer is.

Example

- Node: "MyPet"
- Link: "is-a" (with 0.9 probability)
- Node: "Dog"
- Node: "Dog"
- Link: "has-a" (with 0.95 probability)
- Node: "Tail"
- Node: "Dog"
- Link: "can-do" (with 0.8 probability)
- Node: "Fetch“

- **Question:** Can "MyPet" fetch, and how sure are we?

Cont..

Reasoning:
"MyPet" is a "Dog" with 0.9 certainty.
"Dogs" can "Fetch" with 0.8 certainty.

Estimated Certainty: 0.9 (MyPet is a Dog)×0.8 (Dogs can Fetch)=0.72

So, the AI concludes: "MyPet can Fetch" with **0.72 (72%) certainty.**

Parul® University

NAAC A++ ACCREDITED UNIVERSITY

Probability and Bayes' theorem

- Probability and Bayes' Theorem are fundamental tools for handling **uncertainty** in Artificial Intelligence
- Bayes' theorem is also known as **Bayes' rule, Bayes' law**, or **Bayesian reasoning**, which determines the probability of an event with uncertain knowledge.
- This is useful in dealing with problems, where there is randomness or unpredictability.
- In probability theory, it relates the conditional probability and marginal probabilities of two random events.
- Bayes' theorem was named after the British mathematician **Thomas Bayes**. The **Bayesian inference** is an application of Bayes' theorem, which is fundamental to Bayesian statistics.

Cont..

1. Probability (The Likelihood of Things)

Probability is just a way to measure **how likely something is to happen.** It's expressed as a number between 0 and 1.

- **0:** Means it's impossible. (e.g., Probability of the sun rising from the west = 0)
- **1:** Means it's absolutely certain. (e.g,Probability of the sun rising tomorrow = 1)
- **0.5:** Means it's equally likely to happen or not happen (e.g., Probability of getting heads on a coin flip = 0.5)

Example: If we say:

The probability of a person having a fever is 0.1 (or 10%).

The probability of a person having a cough is 0.2 (or 20%).

These are simple probabilities of individual events(**marginal probability**).

Cont..

- Marginal probability is the probability of an event, irrespective of other random variables.
  - Marginal Probability: The probability of an event irrespective of the outcomes of other random variables, e.g. P(A).
- The joint probability is the probability of two (or more) simultaneous events, often described in terms of events A and B from two dependent random variables, e.g. X and Y. The joint probability is often summarized as just the outcomes, e.g. A and B.
  - Joint Probability: Probability of two (or more) simultaneous events, e.g. P(A and B) or P(A, B).
- The conditional probability is the probability of one event given the occurrence of another event, often described in terms of events A and B from two dependent random variables e.g. X and Y.
  - Conditional Probability: Probability of one (or more) event given the occurrence of another event, e.g. P(A given B) or P(A | B).

Cont..

2. Bayes' Theorem (Updating Your Beliefs with New Evidence)

• Bayes' Theorem is a mathematical rule that tells you **how to update your belief about an event when you get new evidence.**

• It's about figuring out the probability of something *given* that something else has happened.

•Imagine you have an initial belief about something. Then, you observe new information. Bayes' Theorem helps you revise your initial belief to a more accurate one based on this new evidence.

The Formula

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

Cont..

•**P(A|B) (Pronounced "Probability of A given B"):** This is what we want to find out. It's the **posterior probability** – our *updated belief* about event A, after we've seen evidence B.
Example: The probability a patient *has the flu*, *given* they have a fever.

•**P(B|A) (Pronounced "Probability of B given A"):** This is the **likelihood**. It's the probability of seeing the evidence B, *if* A were true. This often comes from data or expert knowledge.
Example: The probability a patient *has a fever*, *if* they truly have the flu.

Cont..

P(A): This is the **prior probability**. It's your initial belief about event A *before* considering any new evidence.

Example: The overall probability of a random person *having the flu* in the general population.

P(B): This is the **evidence probability**. It's the overall probability of seeing the evidence B, regardless of whether A is true or not.

Example: The overall probability of a random person *having a fever* in the general population.

- Where:
- $P(A|B)$ – the probability of event A occurring, given event B has occurred
- $P(B|A)$ – the probability of event B occurring, given event A has occurred
- $P(A)$ – the probability of event A
- $P(B)$ – the probability of event B
- Note that events A and B are independent events

Example: Diagnosing Flu with Bayes' Theorem

We want to know if a patient has the Flu, given they have a Fever.

Given:

P(Flu): The general probability of someone having the Flu (prior belief). Let's say it's **0.05 (5%)** of the population.

P(Fever|Flu): The probability of having a Fever *if* you actually have the Flu. This is common. Let's say it's **0.80 (80%)**.

P(Fever): The general probability of someone having a Fever (can be due to many things like colds, infections, etc.). Let's say it's **0.10 (10%)**.

Question: What is the probability that a patient *has the Flu, given they have a Fever*? (i.e., P(Flu|Fever))

Cont..

Using Bayes' Theorem:

$$P(\text{Flu}|\text{Fever}) = \frac{P(\text{Fever}|\text{Flu}) * P(\text{Flu})}{P(\text{Fever})}$$

$$P(\text{Flu}|\text{Fever}) = \frac{0.80 * 0.05}{0.10}$$

$$P(\text{Flu}|\text{Fever}) = \frac{0.04}{0.10}$$

$$P(\text{Flu}|\text{Fever}) = 0.40$$

Initially, you thought there was only a 5% chance of someone having the Flu. But *after* learning they have a fever, your belief updates. Now, there's a 40% (0.40) chance that the patient has the Flu.

Parul® University

NAAC A++ ACCREDITED UNIVERSITY

Applications of Probability and Bayes' Theorem in AI

- **Medical Diagnosis:** AI systems use it to diagnose diseases based on symptoms and test results.
- **Spam Filtering:** To determine if an email is spam given certain words in it.
- **Robotics:** For robots to update their location or understanding of their environment based on sensor readings.
- **Decision Making:** It helps AI make more informed decisions by quantifying uncertainty and updating beliefs as new data comes in.

In essence, Bayes' Theorem allows AI to learn from data and evidence, becoming "smarter" and more accurate in its predictions and diagnoses over time.

Parul® University

NAAC A++ ACCREDITED UNIVERSITY

Bayesian Network

A Bayesian Network is a powerful probablistic graphical model used in Artificial Intelligence (AI) to represent set of variables and their conditional dependencies via directed acyclic graph(DAG). It's fundamentally built upon the principles of **probability** and **graph theory**.

It combines two key ideas:
Probability: It uses probabilities to quantify how likely certain events are.
Graphs: It uses a visual graph structure to show how different events are related to each other.

Cont..

A Bayesian Network consists of:

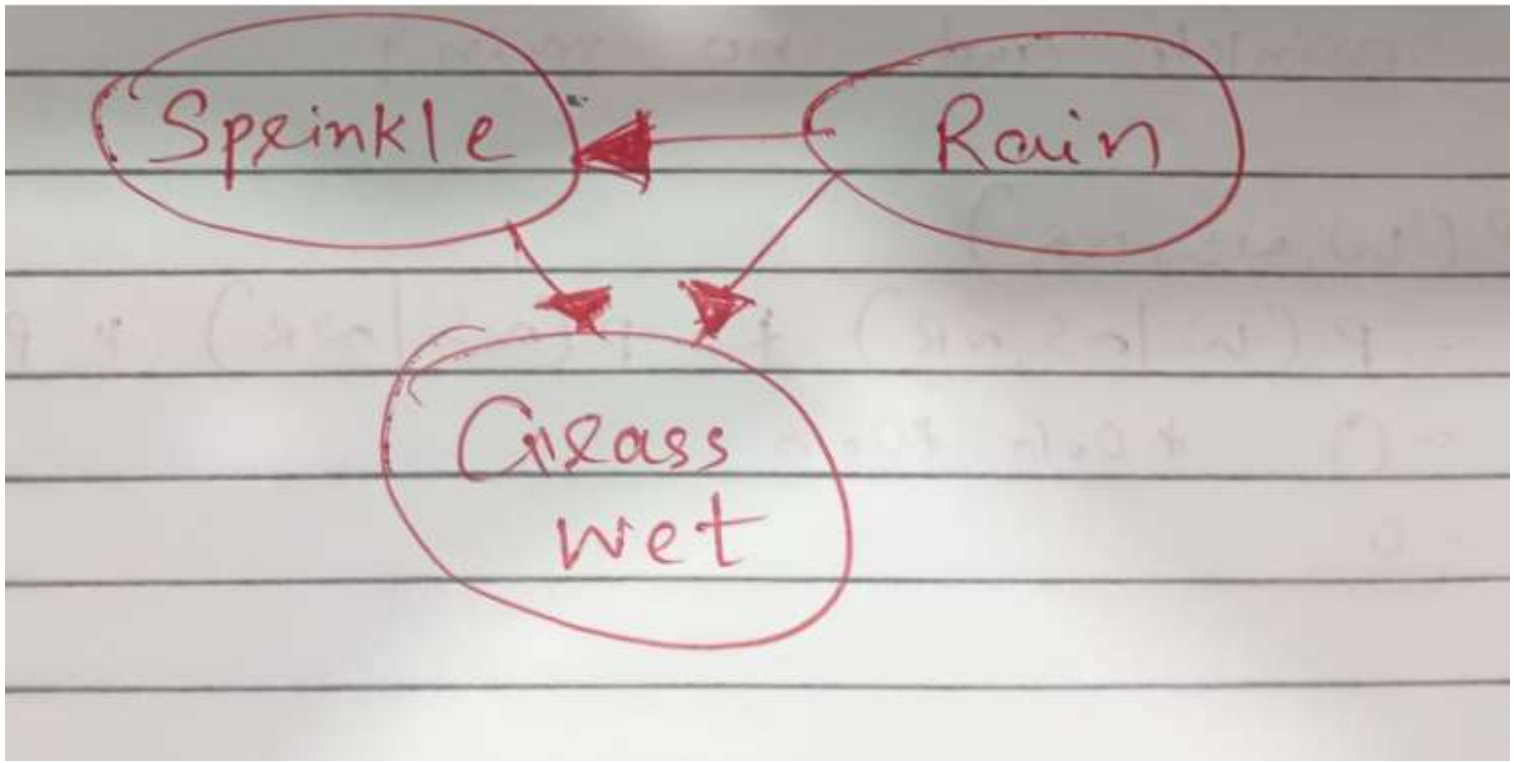
Nodes (Circles): Each node represents a random variable or an event. This could be anything that can be true or false, or take on different values (e.g., "Sky is Cloudy," "Sprinkler is ON," "Grass is Wet," "It Rained").

Directed Edges (Arrows): An arrow from one node to another indicates a **direct causal or influential relationship**. The node at the tail of the arrow is a "parent," and the node at the head is a "child." This means the parent directly influences the child.

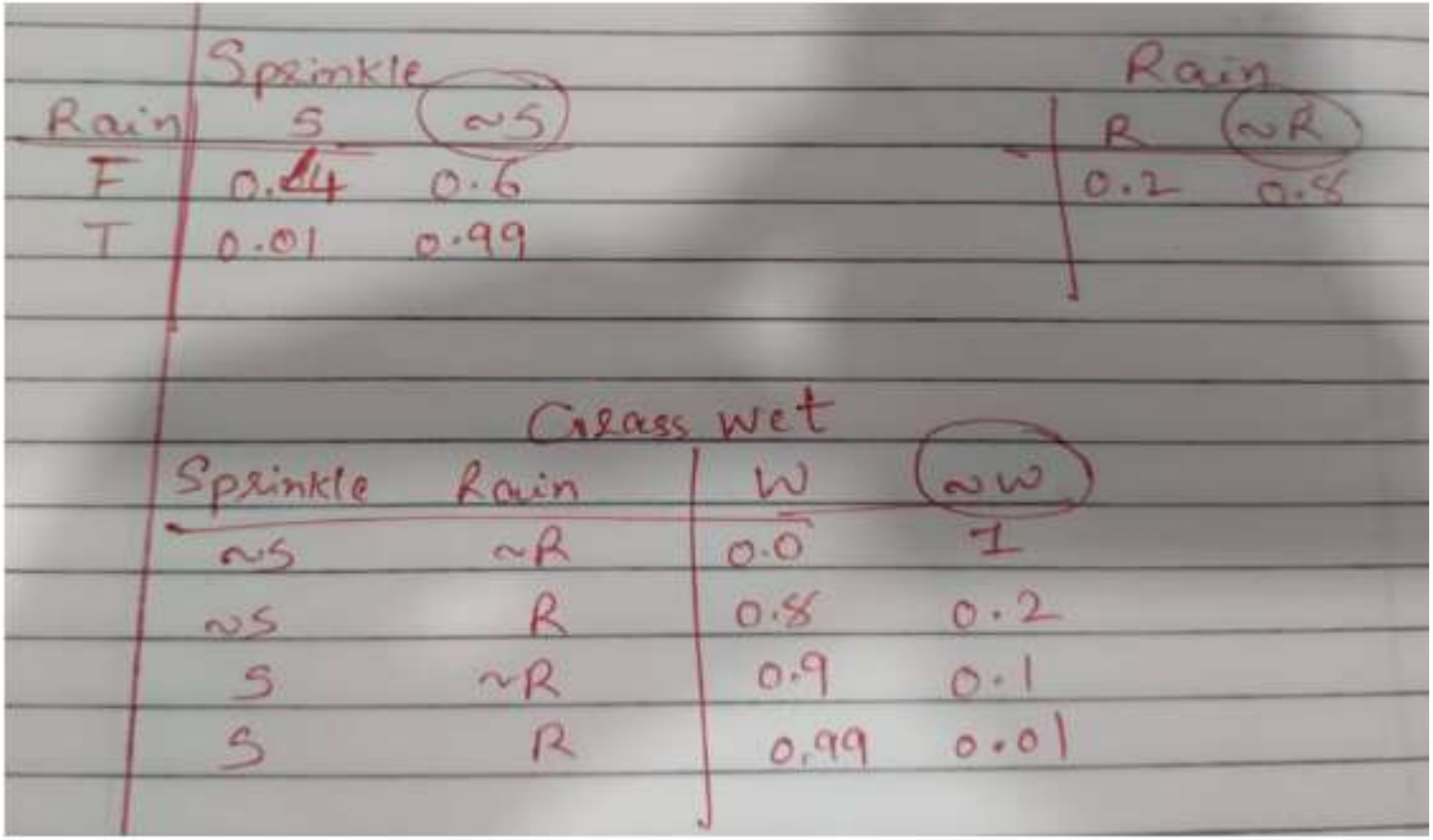
Conditional Probability Tables (CPTs): For each node, there's a table that quantifies the probability of that node being in a certain state, *given the states of its direct parents*. If a node has no parents, its CPT simply lists its prior (initial) probabilities.

Example 1

"Grass is Wet"



Cont..



Rain	Sprinkle S	~S
F	0.4	0.6
T	0.01	0.99

Rain	R	~R
	0.2	0.8

Grass wet

Sprinkle	Rain	W	~W
~S	~R	0.0	1
~S	R	0.8	0.2
S	~R	0.9	0.1
S	R	0.99	0.01

Cont..

Question: What is probability of wet grass when there is a sprinkle and rain?

$$P(W, S, R) = P(W/S,R) * P(S/R) * P(R)$$
$$= 0.99 * 0.01 * 0.2$$
$$= 0.00198$$

Cont..

Question: What is the probability of wet grass, when there is no sprinkle and no rain?

$P(W, \sim S, \sim R)$

$= P(W | \sim S, \sim R) * P(\sim S | \sim R) * P(\sim R)$

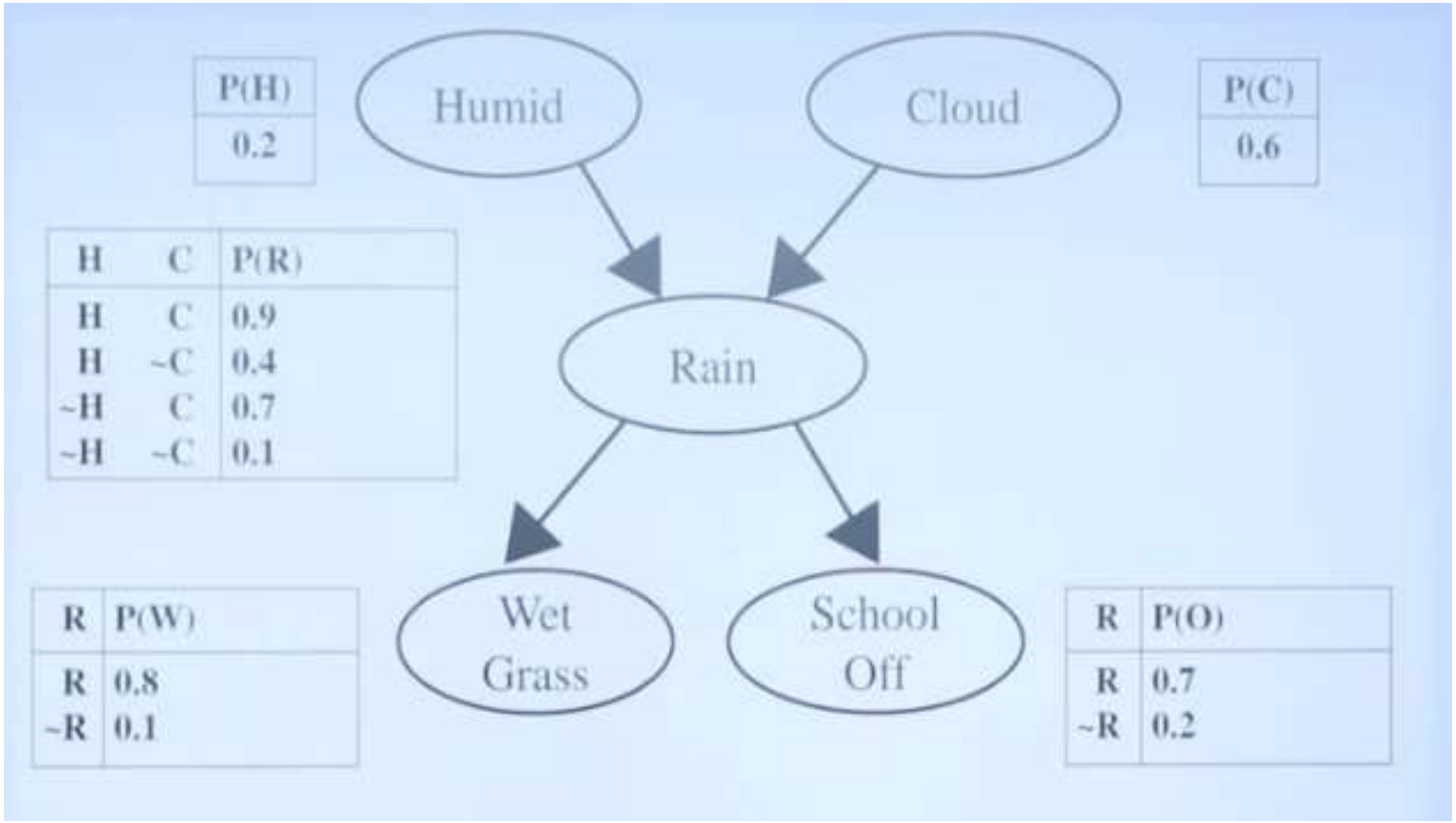
$= 0 * 0.6 * 0.8$

$= 0$

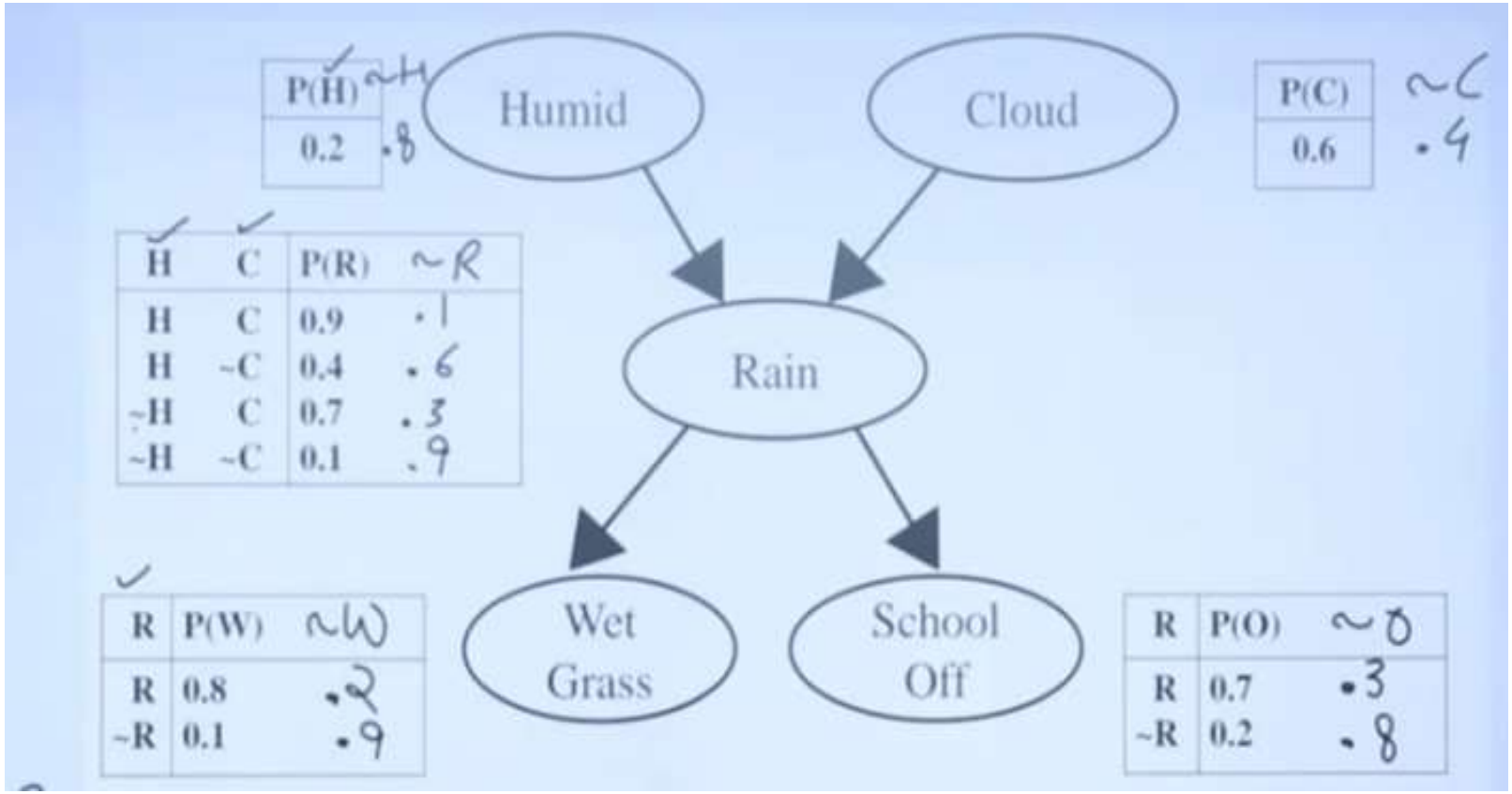
Parul® University

NAAC A++ ACCREDITED UNIVERSITY

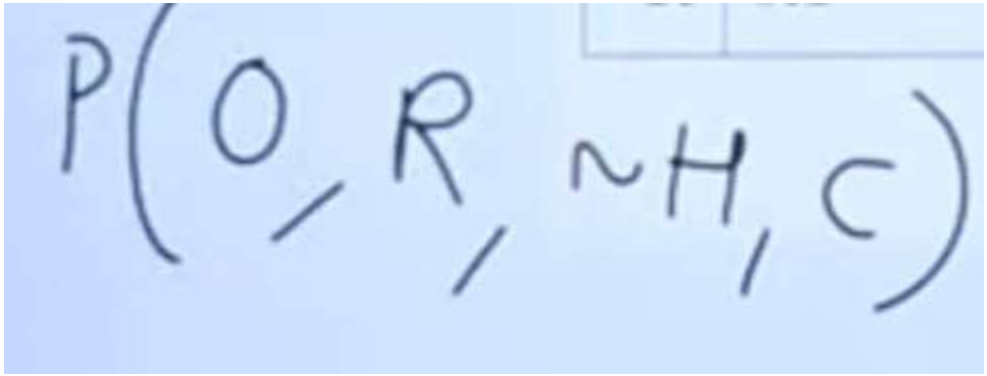
Example 2



Cont..


Humid
Cloud
Rain
Wet Grass
School Off
P(H)
0.2
P(C)
0.6
H C P(R) ~R
H C 0.9 .1
H ~C 0.4 .6
~H C 0.7 .3
~H ~C 0.1 .9
R P(W) ~W
R 0.8 .2
~R 0.1 .9
R P(O) ~O
R 0.7 .3
~R 0.2 .8

Cont..



Cont..

$$P(O, R, \sim H, C) = P(O/R, \sim H, C) \times P(R/\sim H, C) \times P(\sim H) \times P(C)$$

$$= .7 \times .7 \times .8 \times .6$$

Cont..

Nodes (Circles): Each represents an event or variable: "Cloudy," "Rain," "Sprinkler," and "Grass Wet."

Directed Edges (Arrows): Show direct influences. "Cloudy" influences "Rain." Both "Rain" and "Sprinkler" directly influence "Grass Wet."

Conditional Probability Tables (CPTs): These tables define the probabilities of each node's state, given the states of its parents. For example, the "Grass Wet" node would have a CPT detailing the probability of the grass being wet given various combinations of "Rain" and "Sprinkler" being true or false.

Example 3: Medical Diagnosis (Lung Cancer)

we want to determine the likelihood of a person having Lung Cancer based on their smoking habits and whether they have a cough.

Cont..

Nodes:
Smoking: Does the person smoke? (Yes/No)
Lung Cancer: Does the person have lung cancer? (Yes/No)
Cough: Does the person have a cough? (Yes/No)

Directed Edges (Relationships):
Smoking -> Lung Cancer: Smoking is a known risk factor and directly influences the probability of developing lung cancer.
Lung Cancer -> Cough: Lung cancer can directly cause a person to develop a cough.

Conditional Probability Tables (CPTs)
P(Smoking): This would be the prior probability of someone smoking in the general population (e.g., P(Smoking=Yes) = 0.2, P(Smoking=No) = 0.8).
P(Lung Cancer | Smoking):
If Smoking=Yes: P(Lung Cancer=Yes) is higher (e.g., 0.1)
If Smoking=No: P(Lung Cancer=Yes) is lower (e.g., 0.01)

Cont..

P(Cough | Lung Cancer):
If Lung Cancer=Yes: P(Cough=Yes) is higher (e.g., 0.7)
If Lung Cancer=No: P(Cough=Yes) is lower (e.g., 0.1)

An AI system using this network could answer questions like:

•"Given that a person smokes and has a cough, what is the probability they have lung cancer?" (This is crucial for **diagnosis**).
•"If a person has lung cancer, how likely are they to have a cough?" (**Prediction** of symptoms).
•"If a person has a cough, but doesn't smoke, how does that affect the probability of them having lung cancer?" (**Inference** based on partial evidence).

Parul® University

NAAC A++ ACCREDITED UNIVERSITY

Simple Markov Networks(Markov Model)

Markov Network as a way to represent relationships between variables where the influence is mutual or symmetrical, rather than one-way (causal, like in Bayesian Networks). It's a graph where:

•**Nodes (Circles):** Each node represents a random variable or an event.

•**Undirected Edges (Lines without arrows):** A line connecting two nodes means there's a direct *dependency* or *correlation* between them. The influence goes both ways.

Markov networks are extensively used to model complex sequential, spatial, and relational interactions in fields as diverse as image processing, natural language analysis, and bioinformatics.

Markov Property: given the immediate neighbors of a node, that node is conditionally independent of all other nodes in the network. In simpler terms, to know something about a variable, you only need to look at its direct connections, not the whole network.

Cont..

They are particularly useful in:

•**Image Processing:** Modeling relationships between pixels (e.g., neighboring pixels are likely to have similar colors).
•**Computer Vision:** Recognizing objects by modeling dependencies between parts.
•**Social Network Analysis:** Modeling friendships or influences where relationships are reciprocal.

Example 1: Image denoising

We have a noisy black and white image, and you want to clean it up. Each pixel in the image can be either black or white, but some pixels might have been flipped by noise.

Diagram (Conceptual):

```
P1 -- P2 -- P3
|     |     |
P4 -- P5 -- P6
|     |     |
P7 -- P8 -- P9
```

(Where P1, P2, etc., are individual pixels)

Nodes: Each P node represents a single pixel in the image.

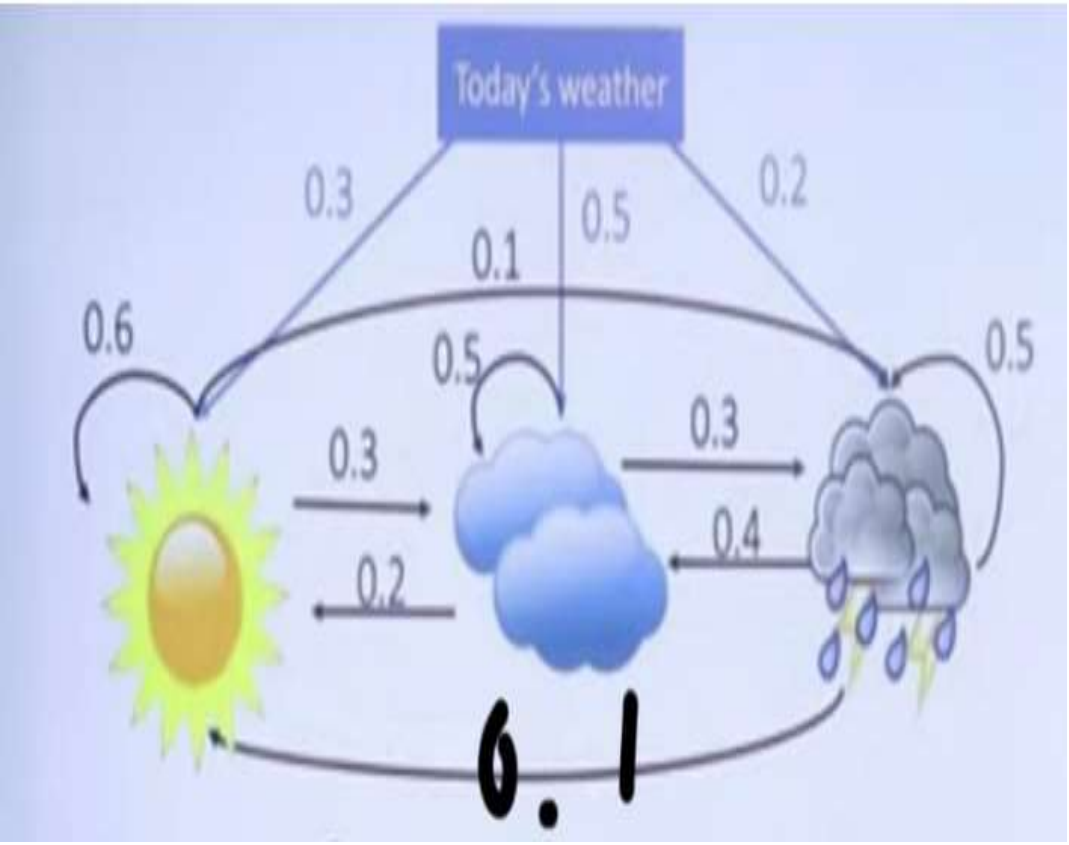
Undirected Edges: A line between two pixels means they are "neighbors." The core idea here is that a pixel's true color is highly dependent on the true colors of its neighboring pixels. For instance, if a pixel's neighbors are all white, it's very likely that this pixel should also be white, even if it appears black due to noise.

Parul® University

NAAC A++ ACCREDITED UNIVERSITY

DIGITAL LEARNING CONTENT

Example 2



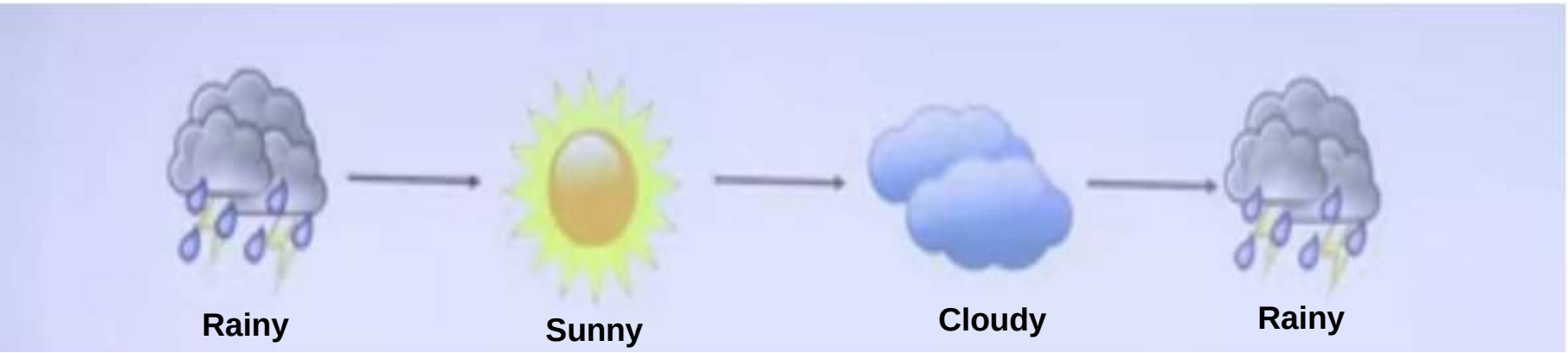
Transition Table

Sunny	Cloudy	Rainy
0.3	0.5	0.2

	Sunny	Cloudy	Rainy
Sunny	0.6	0.3	0.1
Cloudy	0.2	0.5	0.3
Rainy	0.1	0.4	0.5

Parul® University NAAC A++ ACCREDITED UNIVERSITY

Find out the probability of the given sequence.



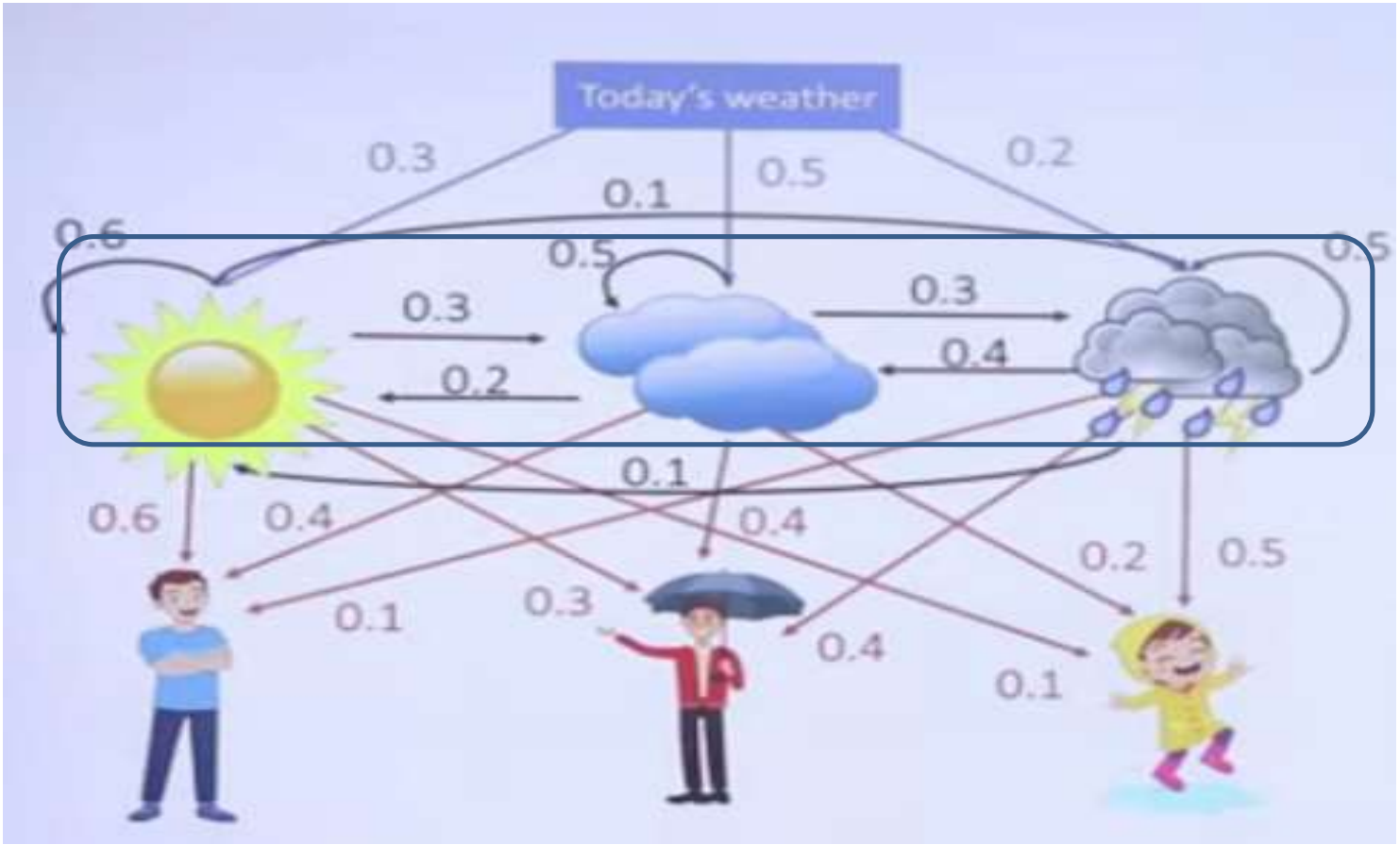
= P(Rainy) * P(Sunny/Rainy) * P(Cloudy/Sunny) * P(Rainy/Cloudy)
=0.2 * 0.1 * 0.3 * 0.3
=0.0018

Question: **Sunny -> Rainy -> Cloudy -> Sunny**

Parul® University

NAAC A++ ACCREDITED UNIVERSITY

Hidden Markov Model



Transition table

Sunny	Cloudy	Rainy
0.3	0.5	0.2

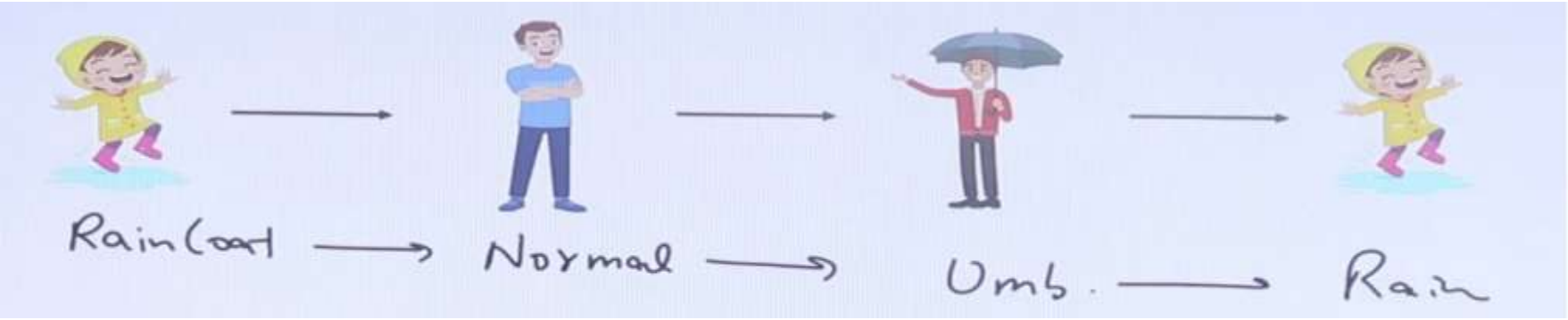
	Sunny	Cloudy	Rainy
Sunny	0.6	0.3	0.1
Cloudy	0.2	0.5	0.3
Rainy	0.1	0.4	0.5

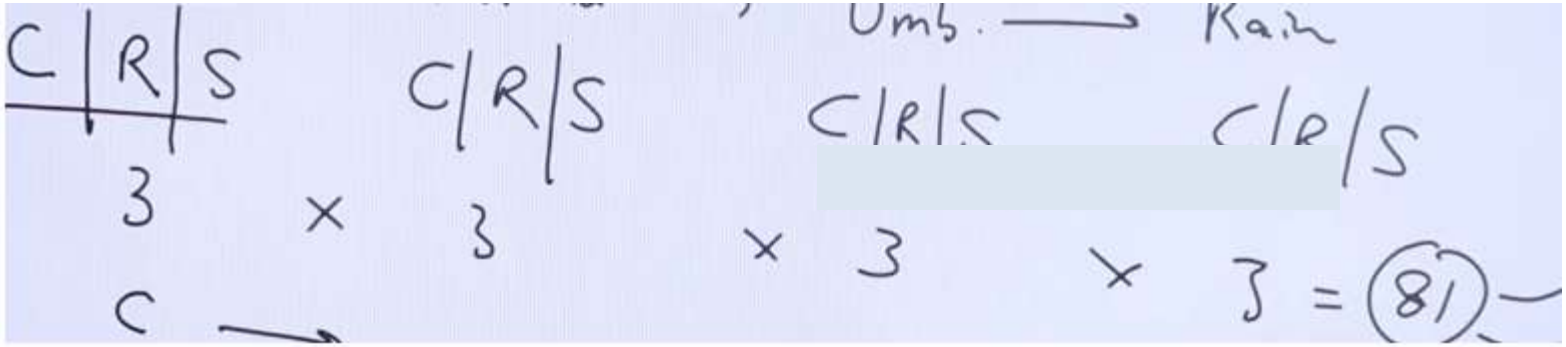
	Normal	Umbrella	Raincoat
Sunny	0.6	0.3	0.1
Cloudy	0.4	0.4	0.2
Rainy	0.1	0.4	0.5

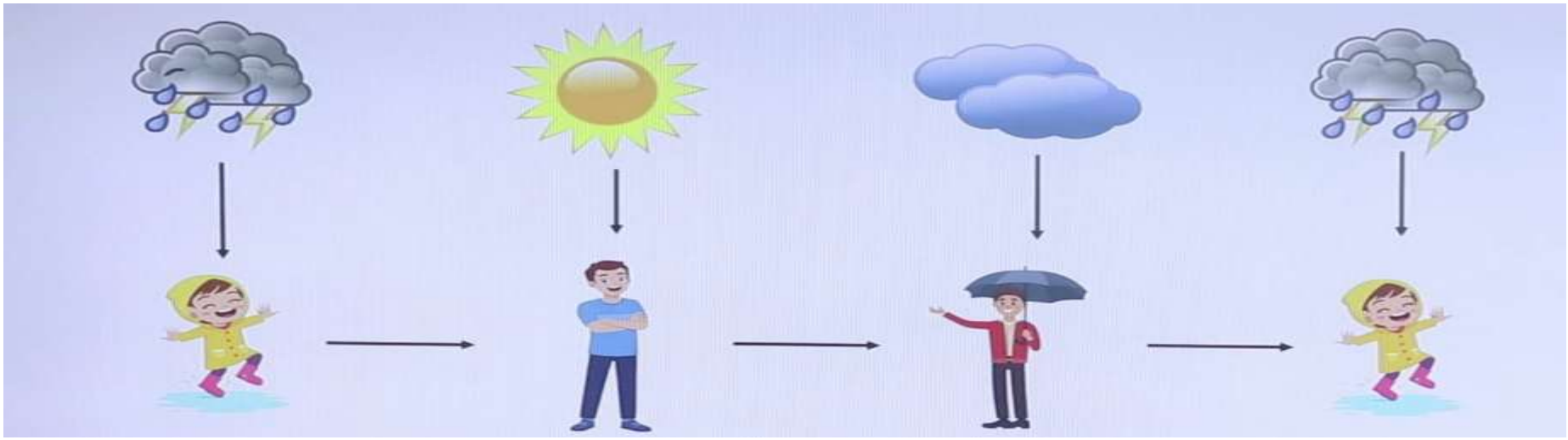
Parul® University

NAAC A++ ACCREDITED UNIVERSITY

Find out the probability of given sequence







= P(Raincoat/Rainy) * P(Normal/Sunny) * P(Umbrella/Cloudy) * P(raincoat/Rainy) * **P(Rainy) *** P(Sunny/Rainy) * P(Cloudy/Sunny) * P(Rainy/Cloudy)

Parul® University NAAC A++ ACCREDITED UNIVERSITY

Hidden Markov Model

- A statistical model called a Hidden Markov Model (HMM) is used to describe systems with changing unobservable states over time.

Imagine Bobby, goes to a casino and calls you every night to tell you whether he won or lost money. You know that the casino has two types of dice:

•**Fair Dice:** Rolls each number (1-6) with equal probability (1/6).

•**Loaded Dice:** Rolls a 6 with a much higher probability (e.g., 0.5) and other numbers (1-5) with a lower probability (e.g., 0.1 each).

Cont..

The casino owner might switch between the fair and loaded dice at any point. Bobby doesn't tell you which dice he's using; he only tells you the outcome of his rolls (won or lost, or more specifically, the actual numbers he rolled).

What are the Hidden and Observed Parts?

- **Hidden States:** The type of dice Bobby is currently using (Fair Dice or Loaded Dice). You don't directly observe this.
- **Observed Outputs (Observations):** The sequence of numbers Bobby rolls (e.g., 1, 6, 3, 6, 6, 2...). You directly observe these.

Parul® University NAAC A++ ACCREDITED UNIVERSITY

Cont..

The "Markov" part means that the probability of being in a particular hidden state at any given time depends *only* on the previous hidden state (not on the entire sequence of states before that). This is called the **Markov assumption**.

- Because of their superior ability to capture uncertainty and temporal dependencies, HMMs are used in a wide range of industries, including finance, bioinformatics, and speech recognition.
- HMMs are useful for modelling dynamic systems and forecasting future states based on sequences that have been seen because of their flexibility.

Cont..

How it Works (Briefly):

An HMM has two main types of variables and probabilities:

•**Hidden States (unobservable):** These are the true, underlying states of the system at each point in time (e.g., a person's mood, the word being spoken, the stock market trend).

•**Observations/Emissions (observable):** These are the data points you actually see (e.g., facial expressions, speech sounds, stock prices).

It's defined by:

•**Transition Probabilities:** The probability of moving from one hidden state to another hidden state (e.g., P(State at t+1 | State at t)).

•**Emission Probabilities:** The probability of observing a particular emission given that the system is in a specific hidden state (e.g., P(Observation | Hidden State)).

•**Initial State Probabilities:** The probability of starting in each hidden state.

Cont..

Initial State Probabilities (π): The probability of starting in each hidden state.
•P(Dice is Fair at start)
•P(Dice is Loaded at start)
Transition Probabilities (A): The probability of moving from one hidden state to another.
•P(Switch to Loaded | Currently Fair)
•P(Stay Fair | Currently Fair)
•P(Switch to Fair | Currently Loaded)
•P(Stay Loaded | Currently Loaded) (e.g., maybe a 0.9 chance of staying with the current dice, and 0.1 chance of switching)
Emission Probabilities (B): The probability of observing a particular output given a hidden state.
•P(Roll a 1 | Dice is Fair)
•P(Roll a 6 | Dice is Fair)
•P(Roll a 1 | Dice is Loaded)
•P(Roll a 6 | Dice is Loaded) (These are based on the probabilities of rolling each number with a fair dice vs. a loaded die as described above).

Parul® University

NAAC A++ ACCREDITED UNIVERSITY

Question

What are the applications of Hidden Markov Models?

Parul® University

NAAC A++ ACCREDITED UNIVERSITY

Answer

Speech recognition, natural language processing, bioinformatics (gene prediction, for example), computer vision, social network analysis and many more fields where systems can be modelled as sequences of observable events with underlying hidden states are applications that heavily rely on HMMs.

Parul® University

NAAC A++ ACCREDITED UNIVERSITY

Basics of Utility Theory

- Artificial intelligence (AI) is now widely used in our daily lives, from smartphone voice assistants to complex decision-making systems in industries like finance and healthcare. ***One fundamental concept that plays a crucial role in decision-making is the utility theory in artificial intelligence.***
- Utility theory in artificial intelligence provides a mathematical framework for understanding how AI systems make choices among different options based on their perceived value or utility.
- **Utility Theory in AI is about formally representing an agent's (or an AI's) preferences for different outcomes, especially when those outcomes are uncertain.** It provides a way to quantify what an AI "values" and how it should make choices to maximize its "happiness" or "reward" over the long run, even when faced with risk.

Cont..

In the real world, AI systems often have to make decisions where the results are not guaranteed. For instance:

•A self-driving car deciding whether to brake hard (risk of rear-end collision) or swerve (risk of hitting an obstacle).

•A medical AI recommending a treatment (risk of side effects vs. benefit of cure).

•A financial AI choosing investments (risk of loss vs. potential for gain).

Utility theory gives AI a principled way to:

•**Assign value:** Put a numerical "score" on how good or bad different outcomes are *to the AI*.

•**Handle risk:** Incorporate the probabilities of different outcomes happening.

•**Make rational decisions:** Choose the action that is most likely to lead to the best overall outcome, considering both value and uncertainty.

Parul® University

NAAC A++ ACCREDITED UNIVERSITY

Cont..

Key Concepts:

1)Utility Function (U(outcome)): This is a mathematical function that assigns a numerical value (a "utility score") to each possible outcome. The higher the utility score, the more desirable the outcome is to the AI.

2)Expected Utility (EU(action)): When an action has uncertain outcomes, we calculate its expected utility. This is the average utility of all possible outcomes, weighted by their probabilities.

3)Rational Agent: In AI, a "rational agent" is one that chooses the action that maximizes its *expected utility*.

Parul® University

NAAC A++ ACCREDITED UNIVERSITY

Example: Investment Decision

There is a small investment. It has two options for a new investment:

Option A: Risky Stock

Outcome 1 (Success): Stock goes up, gain 100$. Probability = 0.6 (60%)

Outcome 2 (Failure): Stock goes down, lose 50$. Probability = 0.4 (40%)

Option B: Safe Bond

Outcome 1 (Success): Bond gives a guaranteed gain of 10$. Probability = 1.0 (100%)

Let's define a simple Utility Function (e.g., utility is just the money gained/lost):

$U(100)=100

$U(10)=10

$U(-50)=−50

Example: Investment Decision

Now, let's calculate the Expected Utility for each option:

Expected Utility of Risky Stock:

$EU(Risky Stock) = (0.6 * U($100)) + (0.4 * U(-50))

EU(RiskyStock)=(0.6times100)+(0.4times−50)

EU(RiskyStock)=60−20=**40**

Expected Utility of Safe Bond:

$EU(Safe Bond) = (1.0 * U(10))

EU(**Safe**Bond)=1.0times10=**10**

Decision: Since **EU(RiskyStock)=40** is greater than EU(SafeBond)=10, a rational AI, based on this utility function, would choose to **invest in the Risky Stock**.

Parul® University

NAAC A++ ACCREDITED UNIVERSITY

Utility Function

- **Utility Function** is a mathematical tool that assigns a numerical "score" or "value" to different outcomes. It's how we quantify how "good" or "desirable" something is.
- Utility functions play a crucial role in AI decision-making by allowing the system to compare and rank different options based on their utility values. AI systems aim to maximize the expected utility, considering the probability of different outcomes occurring.

Properties of Utility Functions:

1)Monotonicity (More is Better): Generally, more of a good thing (like money) leads to higher utility. If Outcome A is strictly preferred to Outcome B, then $U(A)>U(B)$.

2)Transitivity: If you prefer A over B, and B over C, then you must prefer A over C.

Parul University

DIGITAL LEARNING CONTENT

Parul® University

www.paruluniversity.ac.in