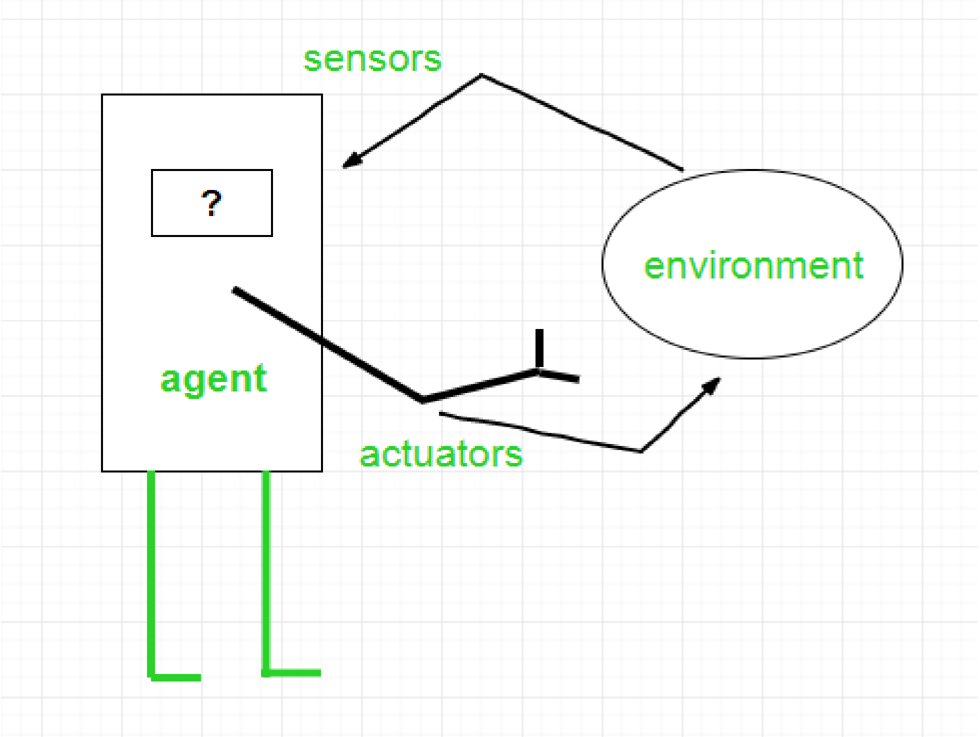
Agent in AI

- Agent in AI :- an agent is a computer program or system that is designed to perceive its environment, make decisions and take actions to achieve a specific goal or set of goals



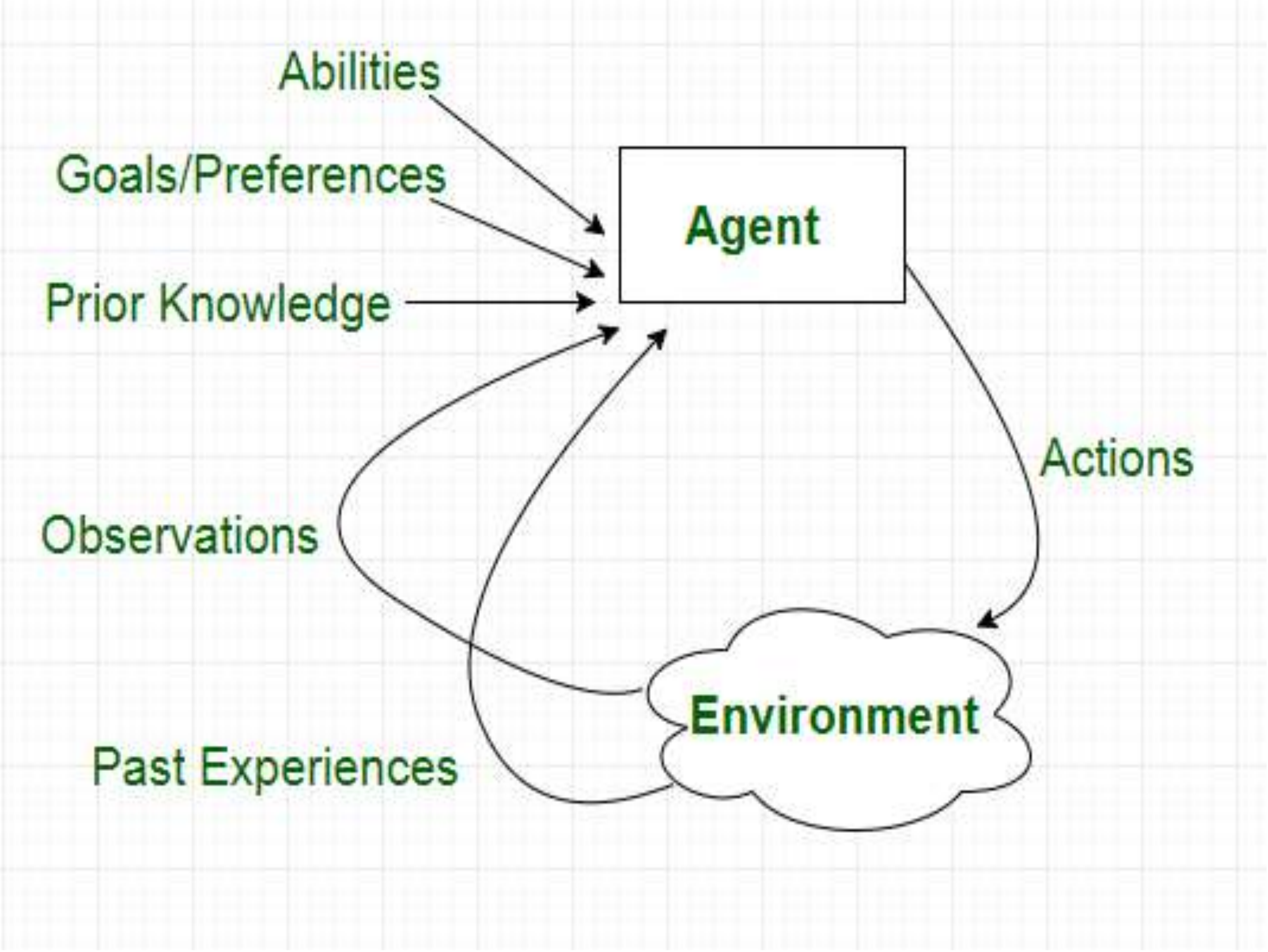
Structure of an AI Agent :-

AI Agent = Architecture + Agent Program

Architecture is the machinery that the agent executes on. It is a device with sensors and actuators, for example, a robotic car, a camera, and a PC.

An agent program is an implementation of an agent function. An **agent function** is a map from the percept sequence(history of all that an agent has perceived to date) to an action.

Characteristics of Agent :-



Search Techniques/Search Algorithm

Search algorithms are one of the most important areas of Artificial Intelligence.

Search Algorithm Terminologies:

- **Search:** Searchingis a step by step procedure to solve a search-problem in a given search space. A search problem can have three main factors:
  - **Search Space:** Search space represents a set of possible solutions, which a system may have.
  - **Start State:** It is a state from where agent begins **the search**.
  - **Goal test:** It is a function which observe the current state and returns whether the goal state is achieved or not.
- **Search tree:** A tree representation of search problem is called Search tree. The root of the search tree is the root node which is corresponding to the initial state.
- **Actions:** It gives the description of all the available actions to the agent.
- **Transition model:** A description of what each action do, can be represented as a transition model.
- **Path Cost:** It is a function which assigns a numeric cost to each path.
- **Solution:** It is an action sequence which leads from the start node to the goal node.
- **Optimal Solution:** If a solution has the lowest cost among all solutions.

Properties of Search Algorithms:

1> **Completeness:** A search algorithm is said to be complete if it guarantees to return a solution if at least any solution exists for any random input.

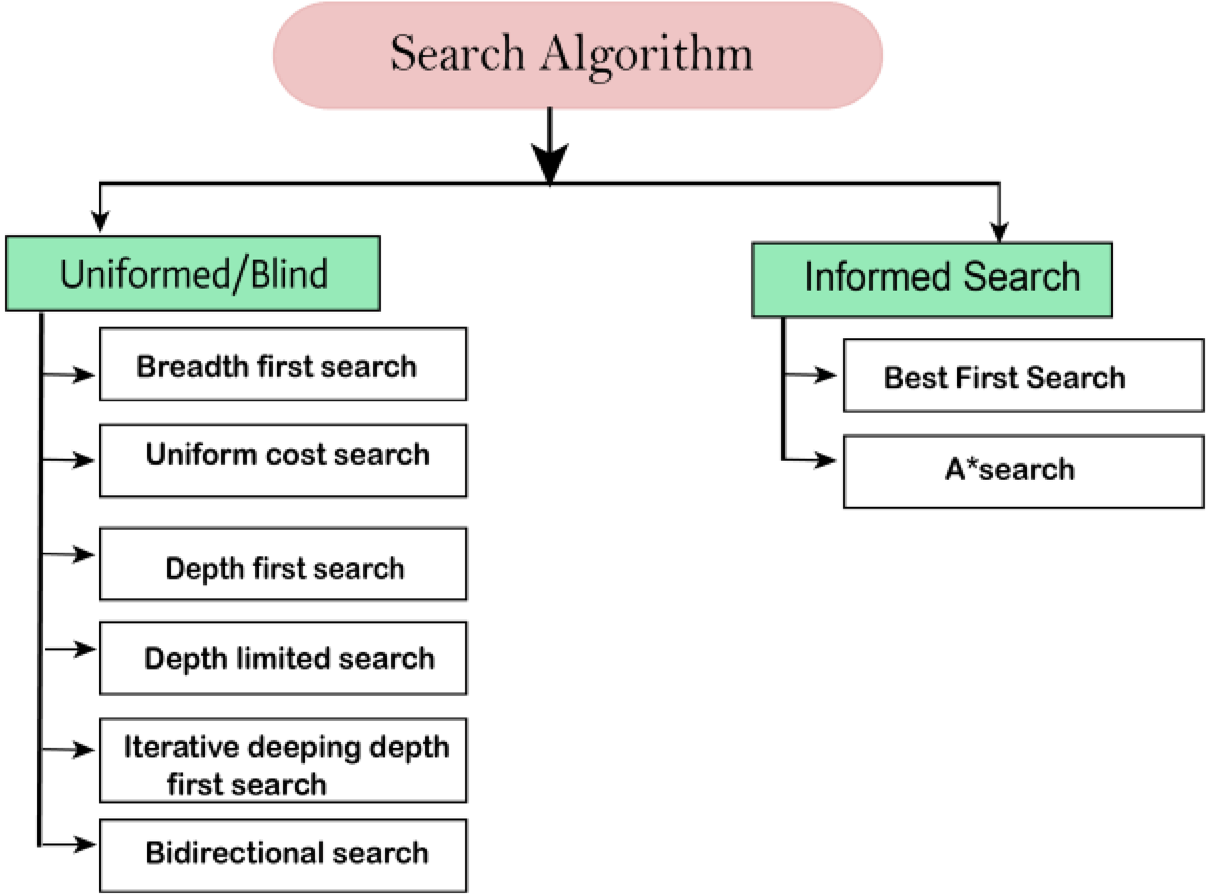
2> **Optimality:** If a solution found for an algorithm is guaranteed to be the best solution (lowest path cost) among all other solutions, then such a solution for is said to be an optimal solution.

3> **Time Complexity:** Time complexity is a measure of time for an algorithm to complete its task.

4>**Space Complexity:** It is the maximum storage space required at any point during the search, as the complexity of the problem

Types of search algorithms

Based on the search problems we can classify the search algorithms into uninformed (Blind search) search and informed search (Heuristic search) algorithms.



Search Algorithm

Uniformed/Blind

Breadth first search

Uniform cost search

Depth first search

Depth limited search

Iterative deeping depth first search

Bidirectional search

Informed Search

Best First Search

A*search

Uninformed/Blind Search

Uninformed/Blind Search:-

1)The uninformed search does not contain any domain knowledge

2) It operates in a brute-force way

- **It can be divided into five main types:**
- Breadth-first search
- Uniform cost search
- Depth-first search
- Iterative deepening depth-first search
- Bidirectional Search

Informed Search

Informed Search

Informed search algorithms use domain knowledge

In an informed search, problem information is available which can guide the search.

Informed search strategies can find a solution more efficiently than an uninformed search strategy

- An example of informed search algorithms is a traveling salesman problem.
- Greedy Search
- A* Search-path finding and graph traversal.

- Depth-first search (DFS) is an algorithm for traversing or searching tree or graph data structures. The algorithm starts at the root node (selecting some arbitrary node as the root node in the case of a graph) and explores as far as possible along each branch before backtracking. It uses last in-first-out strategy and hence it is implemented using a stack.
- Breadth-first search (BFS) is an algorithm for traversing or searching tree or graph data structures. It starts at the tree root (or some arbitrary node of a graph, sometimes referred to as a ‘search key’), and explores all of the neighbor nodes at the present depth prior to moving on to the nodes at the next depth level. It is implemented using a queue.
- UCS is different from BFS and DFS because here the costs come into play. In other words, traversing via different edges might not have the same cost. The goal is to find a path where the cumulative sum of costs is the least.

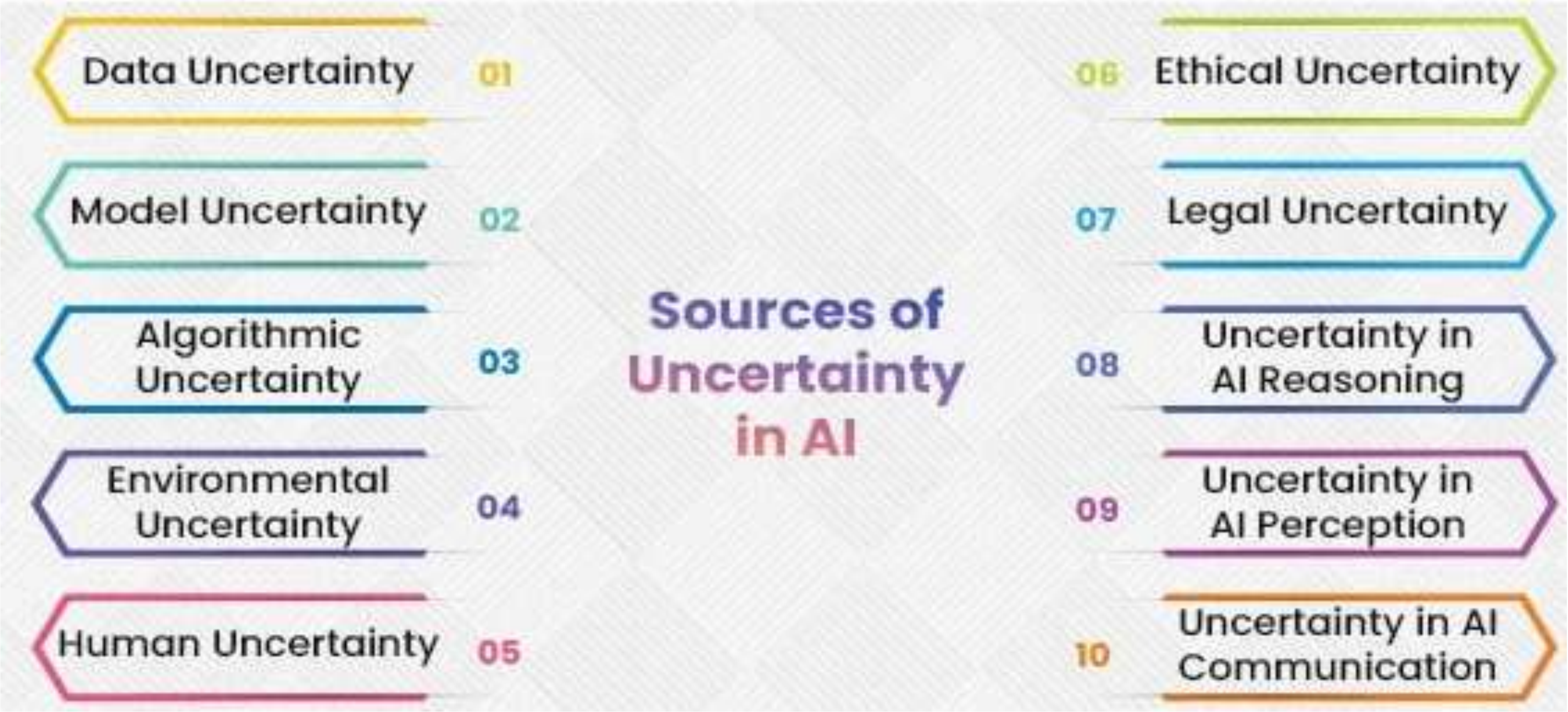
- **Greedy Search:**
- In greedy search, we expand the node closest to the goal node. The “closeness” is estimated by a heuristic h(x). **Heuristic:** A heuristic h is defined as-
  h(x) = Estimate of distance of node x from the goal node.
  Lower the value of h(x), closer is the node from the goal.

- A* Tree Search, or simply known as A* Search, combines the strengths of uniform-cost search and greedy search. In this search, the heuristic is the summation of the cost in UCS, denoted by g(x), and the cost in the greedy search, denoted by h(x). The summed cost is denoted by f(x).

  Heuristic: The following points should be noted wrt heuristics in A* search.
- Here, h(x) is called the **forward cost** and is an estimate of the distance of the current node from the goal node.
- And, g(x) is called the **backward cost** and is the cumulative cost of a node from the root node.
- A* search is optimal only when for all nodes, the forward cost for a node h(x) underestimates the actual cost h*(x) to reach the goal. This property of *A** heuristic is called **admissibility**.

Uncertainty in AI….????

- Artificial intelligence (AI) uncertainty is when there’s not enough information or ambiguity in data or decision-making. It is a fundamental concept in AI, as real-world data is often noisy and incomplete.
- AI systems must account for uncertainty to make informed decisions.
- 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 in AI

01 Data Uncertainty

02 Model Uncertainty

03 Algorithmic Uncertainty

04 Environmental Uncertainty

05 Human Uncertainty

06 Ethical Uncertainty

07 Legal Uncertainty

08 Uncertainty in AI Reasoning

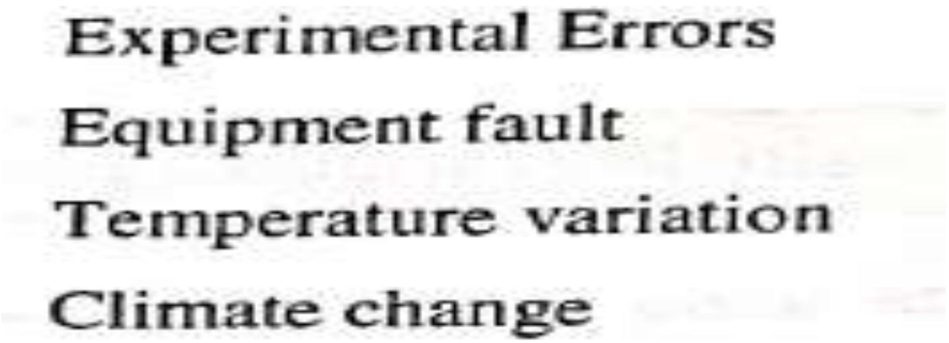
09 Uncertainty in AI Perception

10 Uncertainty in AI Communication

- **Data Uncertainty:** AI models are trained on data, and the quality and accuracy of the data can affect the performance of the model. Noisy or incomplete data can lead to uncertain predictions or decisions made by the AI system.
- **Model Uncertainty:** AI models are complex and can have various parameters and hyper parameters that need to be tuned. The choice of model architecture, optimization algorithm, and hyper parameters can significantly impact the performance of the model, leading to uncertainty in the results.
- **Algorithmic Uncertainty:** AI algorithms can be based on different mathematical formulations, leading to different results for the same problem. For example, different machine learning algorithms can produce different predictions for the same dataset.
- **Environmental Uncertainty:** AI systems operate in dynamic environments, and changes in the environment can affect the performance of the system. For example, an autonomous vehicle may encounter unexpected weather conditions or road construction that can impact its ability to navigate safely.
- **Human Uncertainty:** AI systems often interact with humans, either as users or as part of the decision-making process. Human behavior and preferences can be difficult to predict, leading to uncertainty in the use and adoption of AI systems.

- **Ethical Uncertainty:** AI systems often raise ethical concerns, such as privacy, bias, and transparency. These concerns can lead to uncertainty in the development and deployment of AI systems, particularly in regulated industries.
- **Legal Uncertainty:** AI systems must comply with laws and regulations, which can be ambiguous or unclear. Legal challenges and disputes can arise from the use of AI systems, leading to uncertainty in their adoption and implementation.
- **Uncertainty in AI Reasoning:** AI systems use reasoning techniques to make decisions or predictions. However, these reasoning techniques can be uncertain due to the complexity of the problems they address or the limitations of the data used to train the models.
- **Uncertainty in AI Perception:** AI systems perceive their environment through sensors and cameras, which can be subject to noise, occlusion, or other forms of interference. This can lead to uncertainty in the accuracy of the data used to train AI models or the effectiveness of AI systems in real-world applications.
- **Uncertainty in AI Communication:** AI systems communicate with humans through natural language processing or computer vision. However, language and visual cues can be ambiguous or misunderstood, leading to uncertainty in the effective communication between humans and AI systems.

- What are the causes of uncertainty?
- **The information occurred from unreliable sources :**
- Missing information. We can be uncertain because we are missing important information. ...
- Unreliable information. We can be uncertain because we aren't able to trust the information, even if we have it. ...
- Conflicting information. ...
- Noisy information. ...
- Confusing information.



Experimental Errors

Equipment fault

Temperature variation

Climate change

- Probability plays a central role in AI by providing a formal framework for handling uncertainty.
- AI systems use probabilistic models and reasoning to make informed decisions, assess risk, and quantify uncertainty, allowing them to operate effectively in complex and uncertain real-world scenarios.
- Probability(Event) = Favorable Outcomes/Total Outcomes = x/n
- In probability, there are two ways to solve problems when we’re not sure about the information:
- Bayes’ rule
- Bayesian statistics
- making decisions with incomplete or unclear information.

- **Terminology of Probability Theory**
- The following terms in probability theorey help in a better understanding of the concepts of probability.
- **Experiment:** A trial or an operation conducted to produce an outcome is called an experiment.
- **Sample Space:** All the possible outcomes of an experiment together constitute a sample space. For example, the sample space of tossing a coin is {head, tail}.
- **Favorable Outcome:** An event that has produced the desired result or expected event is called a favorable outcome. For example, when we roll two dice, the possible/favorable outcomes of getting the sum of numbers on the two dice as 4 are (1,3), (2,2), and (3,1).

- **Trial:** A trial denotes doing a random experiment.
- **Random Experiment:** An experiment that has a well-defined set of outcomes is called a random experiment. For example, when we toss a coin, we know that we would get a head or tail, but we are not sure which one will appear.
- **Event:** The total number of outcomes of a random experiment is called an event.
- **Equally Likely Events:** Events that have the same chances or probability of occurring are called equally likely events. The outcome of one event is independent of the other. For example, when we toss a coin, there are equal chances of getting a head or a tail.

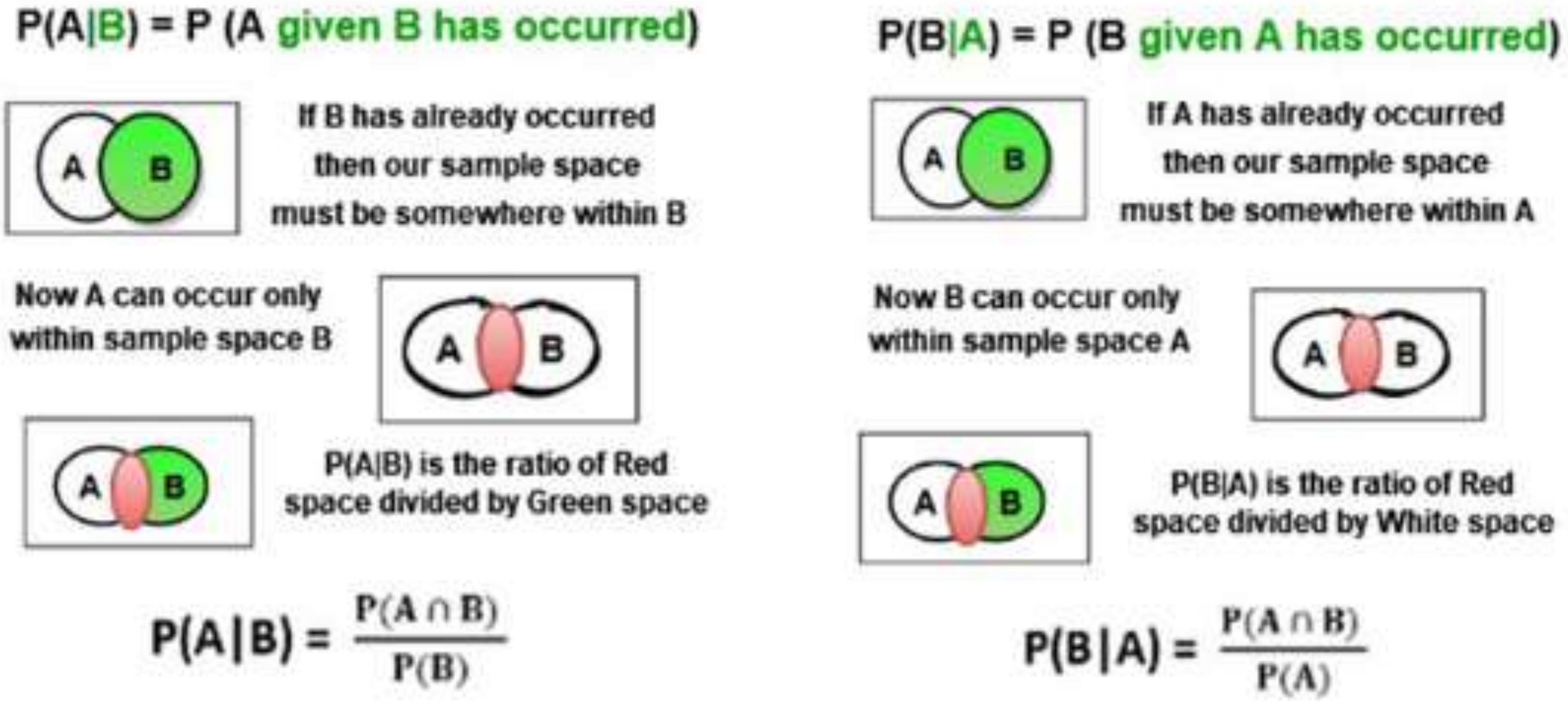
- **Exhaustive Events:** When the set of all outcomes of an event is equal to the sample space, we call it an exhaustive events.
- **Mutually Exclusive Events:** Events that cannot happen simultaneously are called mutually exclusive events. For example, the climate can be either hot or cold. We cannot experience the same weather simultaneously.
- **Events in Probability**
- In probability theory, an event is a set of outcomes of an experiment or a subset of the sample space. If P(E) represents the probability of an event E, then, we have,
- P(E) = 0 if and only if E is an impossible event.
- P(E) = 1 if and only if E is a certain event.
- $0 \leq P(E) \leq 1$.

- **Prior Probability**: This is the probability of an event before any additional evidence is taken into account. It represents what we believe or know about the likelihood of an event happening based on past information or initial assumptions.
- **Posterior Probability**: This is the probability of an event occurring after considering new evidence or information. It's updated based on prior probabilities and the new evidence, using Bayes' theorem or similar methods.
- **Conditional Probability**: This is the probability of an event occurring given that another event has already occurred. It quantifies the likelihood of one event happening under the condition that another event has occurred or is known to have occurred.

- **Bayes' Rule:** for things like sorting things into groups, making guesses about the future, and deciding what to do when things are uncertain.
- **Mathematically, Bayes' theorem is expressed as follows:**
- ***P(A|B) = (P(B|A) * P(A)) / P(B)*** **Here,**
- The posterior probability, represented by **P(A|B)**, is the chance of event A happening when event B has happened.
- **P(B|A)** shows how likely event B is when event A has already happened.
- The prior probability, **P(A)**, is the initial chance of event A happening before any new information is considered.
- **P(B)** is the probability of event B happening, whether or not event A has happened.
- In AI, Bayes' theorem updates probabilities of hypotheses or predictions with new data or evidence. It is helpful for dealing with uncertainty and making decisions with incomplete or unclear information.

- Suppose, we are given two events, "A" and "B", then the probability of event A, $P(A) > P(B)$ if and only if event "A" is more likely to occur than the event "B". Sample space(S) is the set of all of the possible outcomes of an experiment and n(S) represents the number of outcomes in the sample space.
- $P(E) = n(E)/n(S)$
- $P(E') = (n(S) - n(E))/n(S) = 1 - (n(E)/n(S))$
- E' represents that the event will not occur.
- Therefore, now we can also conclude that, $P(E) + P(E') = 1$

- i.e., P(A) = n(A)/n(S)
- where,
- P(A) is the probability of an event 'B'.
- n(A) is the number of favorable outcomes of an event 'B'.
- n(S) is the total number of events occurring in a sample space.



P(A|B) = P (A given B has occurred)

If B has already occurred then our sample space must be somewhere within B

Now A can occur only within sample space B

P(A|B) is the ratio of Red space divided by Green space

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

P(B|A) = P (B given A has occurred)

If A has already occurred then our sample space must be somewhere within A

Now B can occur only within sample space A

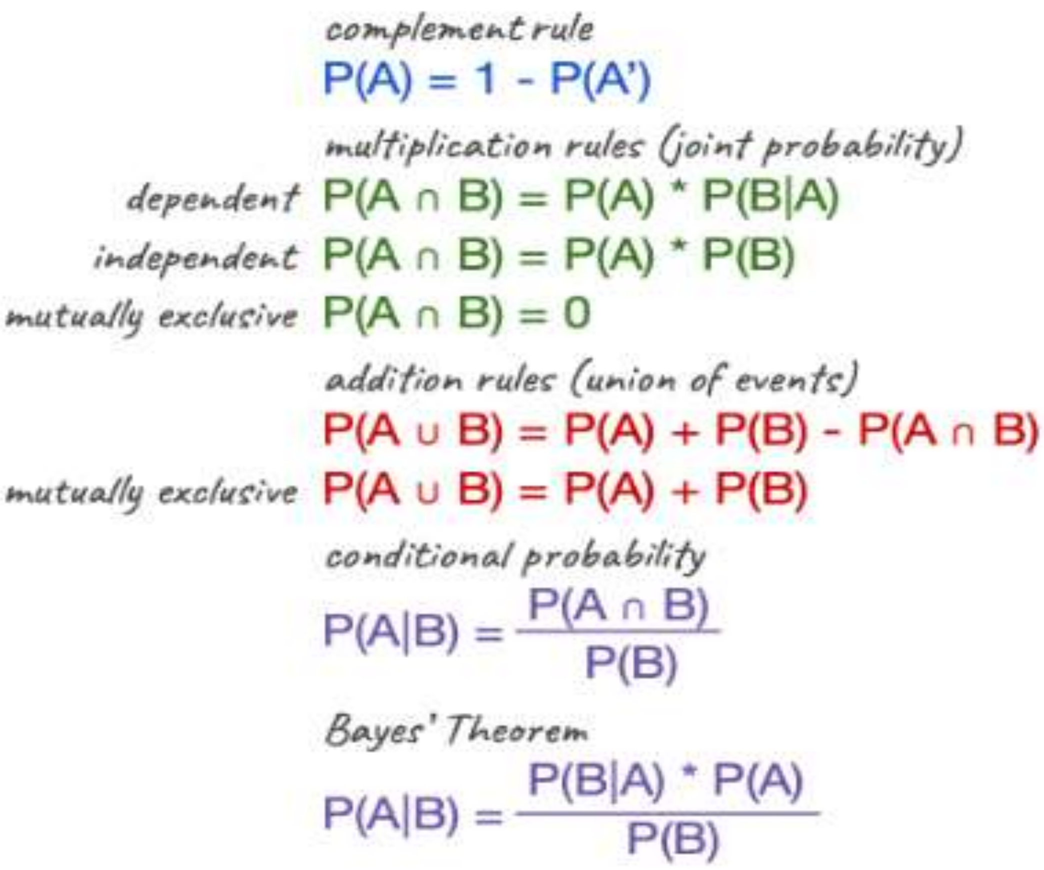
P(B|A) is the ratio of Red space divided by White space

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

For two events A and B, the chain rule states that

$$\mathbb{P}(A \cap B) = \mathbb{P}(B \mid A)\mathbb{P}(A),$$

where $\mathbb{P}(B \mid A)$ denotes the conditional probability of B given A.



complement rule

$P(A) = 1 - P(A')$

multiplication rules (joint probability)

dependent $P(A \cap B) = P(A) * P(B|A)$

independent $P(A \cap B) = P(A) * P(B)$

mutually exclusive $P(A \cap B) = 0$

addition rules (union of events)

$P(A \cup B) = P(A) + P(B) - P(A \cap B)$

mutually exclusive $P(A \cup B) = P(A) + P(B)$

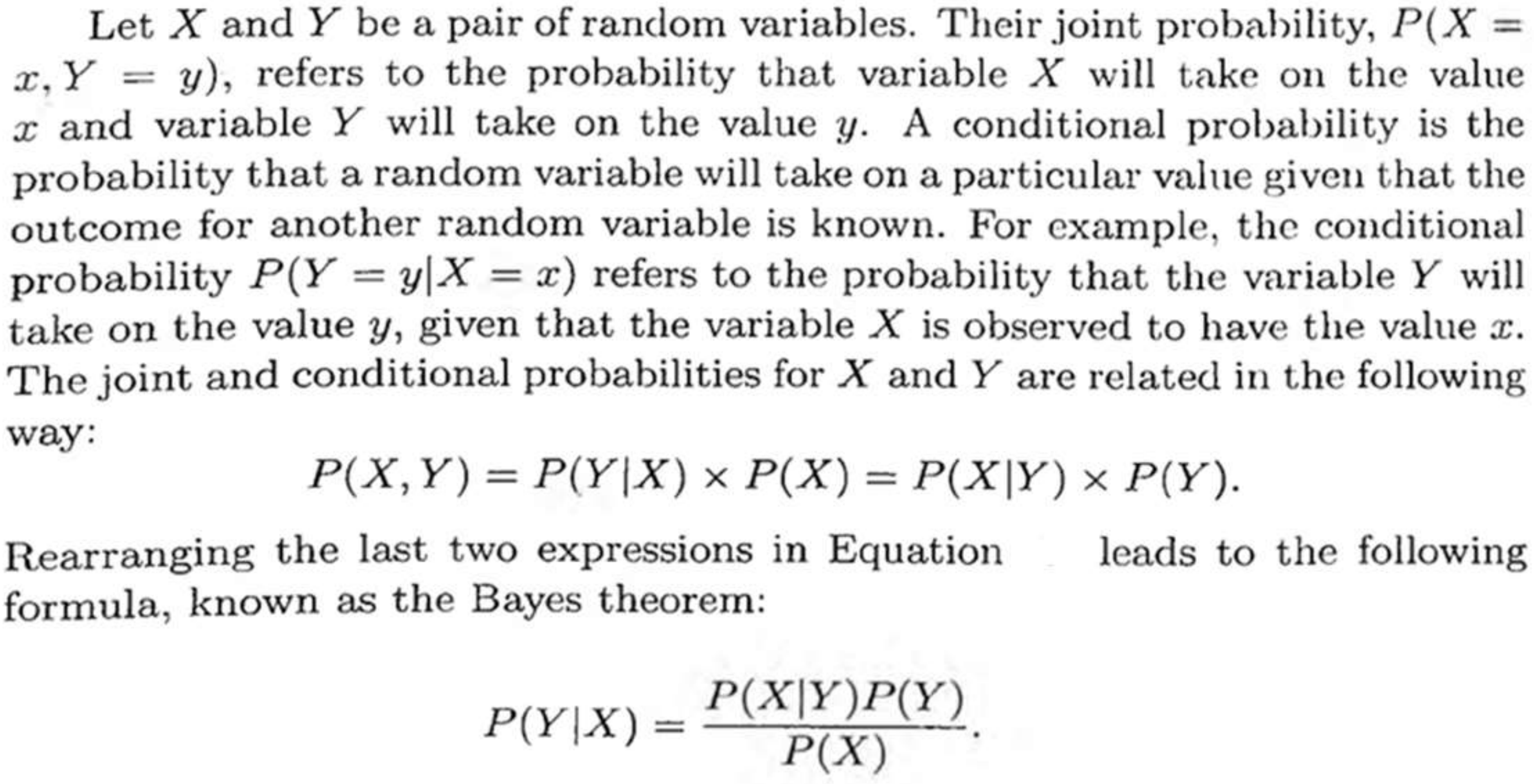
conditional probability

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

Bayes' Theorem

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

the joint probability of events A and B is expressed formally as:
The letter P is the first letter of the alphabet (A and B).
The upside-down capital “U” operator or, in some situations, a comma “,” represents the “and” or conjunction.
P(A ^ B)
P(A, B)

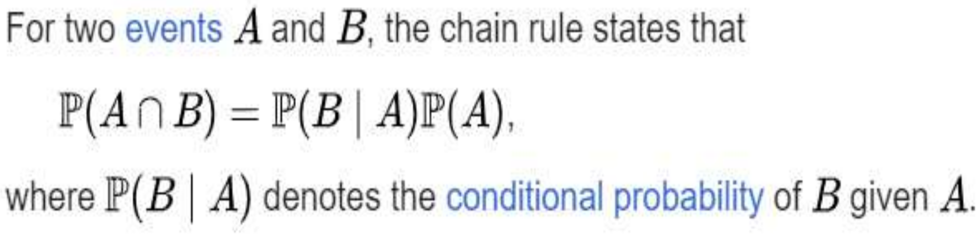


Let X and Y be a pair of random variables. Their joint probability, $P(X = x, Y = y)$, refers to the probability that variable X will take on the value x and variable Y will take on the value y. A conditional probability is the probability that a random variable will take on a particular value given that the outcome for another random variable is known. For example, the conditional probability $P(Y = y|X = x)$ refers to the probability that the variable Y will take on the value y, given that the variable X is observed to have the value x. The joint and conditional probabilities for X and Y are related in the following way:

$$P(X, Y) = P(Y|X) \times P(X) = P(X|Y) \times P(Y).$$

Rearranging the last two expressions in Equation leads to the following formula, known as the Bayes theorem:

$$P(Y|X) = \frac{P(X|Y)P(Y)}{P(X)}.$$



For two events A and B, the chain rule states that

$$\mathbb{P}(A \cap B) = \mathbb{P}(B \mid A)\mathbb{P}(A),$$

where $\mathbb{P}(B \mid A)$ denotes the conditional probability of B given A.

$$P(A \wedge B) = P(A|B)P(B)$$

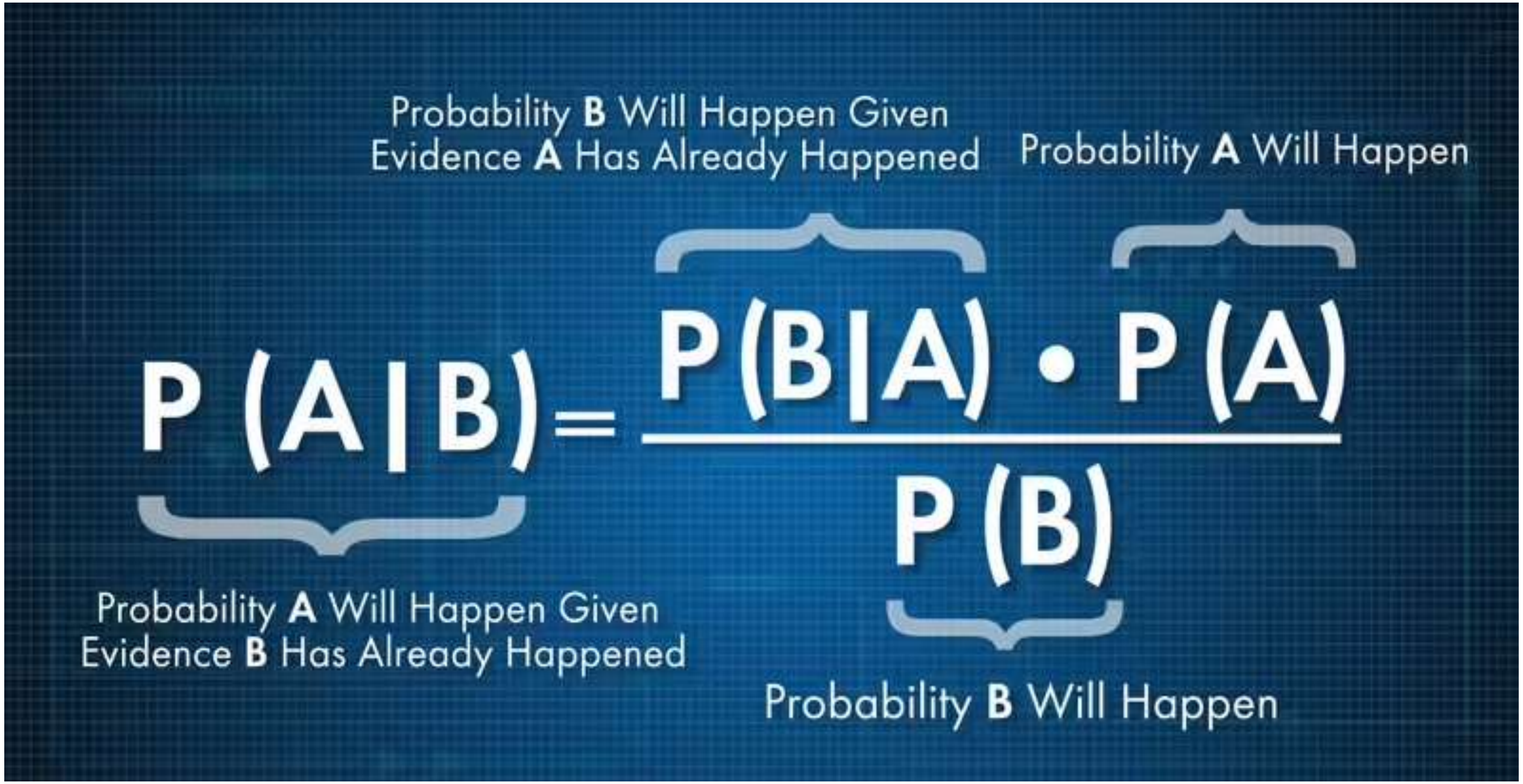
$$P(B \wedge A) = P(B|A)P(A)$$

$$P(B|A)P(A) = P(A|B)P(B)$$

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

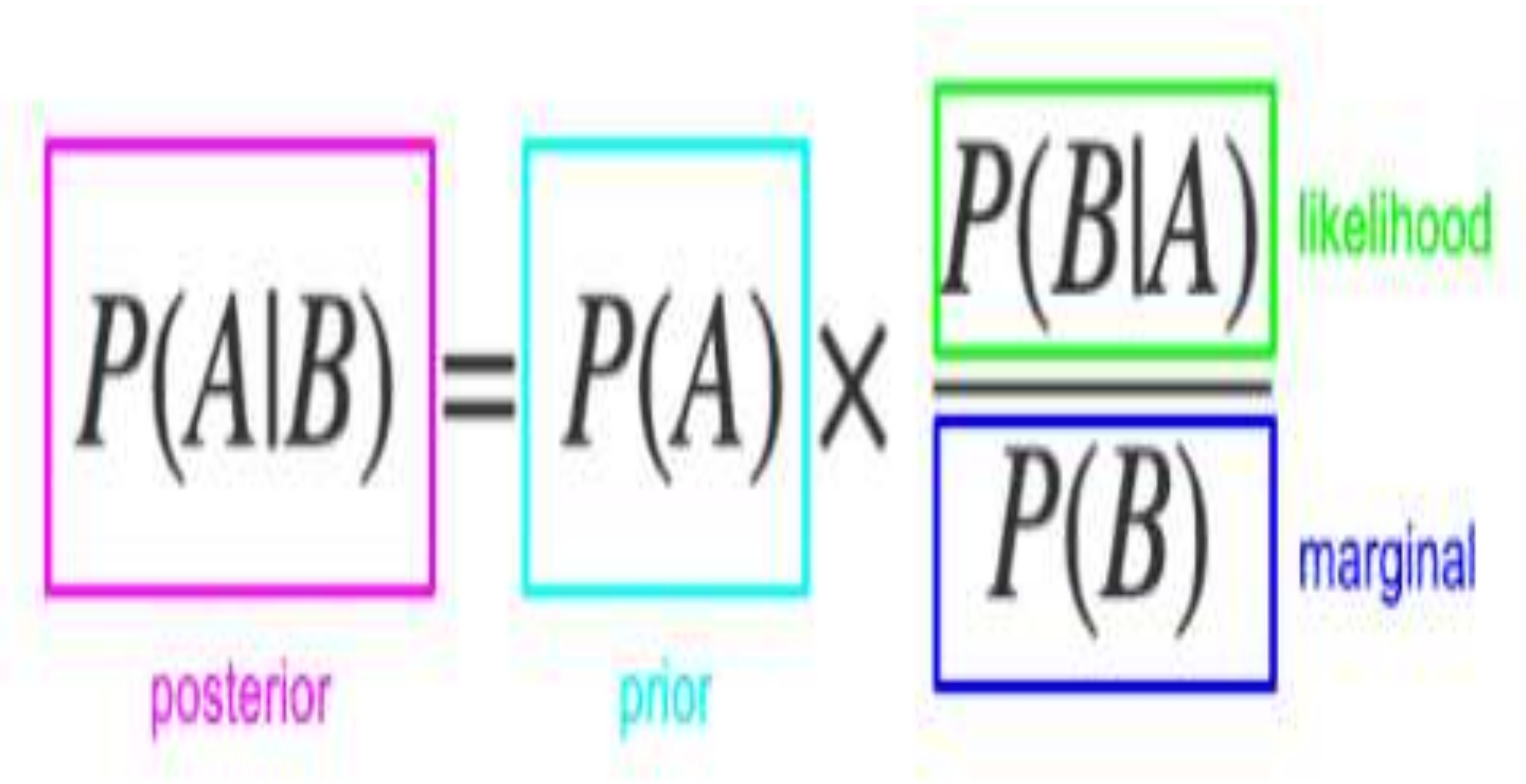
Bayes’ law (also Bayes’ law or Bayes’ rule) is fundamental to probabilistic reasoning in AI!!

Bayes Theorem :- describes the probability of an event, based on prior knowledge of conditions ...



$$\underbrace{P\,(A|B)}_{\substack{\text{Probability } \textbf{A} \text{ Will Happen Given} \\ \text{Evidence } \textbf{B} \text{ Has Already Happened}}} = \frac{\overbrace{P(B|A)}^{\substack{\text{Probability } \textbf{B} \text{ Will Happen Given} \\ \text{Evidence } \textbf{A} \text{ Has Already Happened}}} \cdot \overbrace{P\,(A)}^{\text{Probability } \textbf{A} \text{ Will Happen}}}{\underbrace{P\,(B)}_{\text{Probability } \textbf{B} \text{ Will Happen}}}$$

Bayes Theorem



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

- Bayes’ rule requires three terms - a conditional probability and two unconditional probabilities - just to compute one conditional probability.
- Bayes’ rule is useful in practice because there are many cases where we do have good probability estimates for these three numbers and need to compute the fourth.
- In a task such as medical diagnosis, we often have conditional probabilities on causal relationships and want to derive a diagnosis.

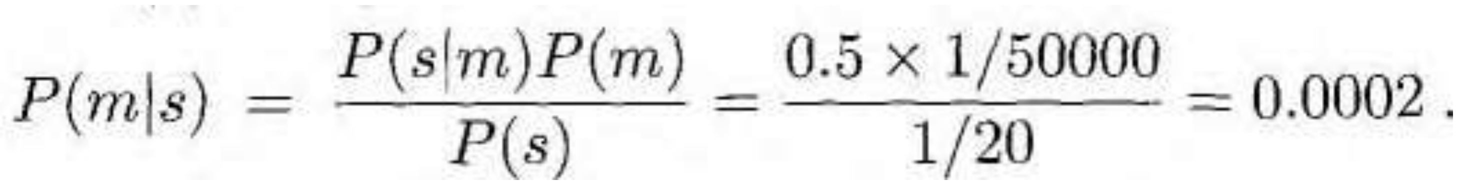
Example:

A doctor knows that the disease meningitis causes the patient to have a stiff neck, say, 50% of the time. The doctor also knows some unconditional facts: the prior probability that a patient has meningitis is 1/50,000, and the prior probability that any patient has a stiff neck is 1/20. Let *s be the proposition that the patient has a stiff neck and m be the proposition that the patient has meningitis.*

$$P(s|m) = 0.5$$

$$P(m) = 1/50000$$

$$P(s) = 1/20$$



$$P(m|s) \;=\; \frac{P(s|m)P(m)}{P(s)} = \frac{0.5 \times 1/50000}{1/20} = 0.0002\,.$$

Consider a football game between two rival teams: Team 0 and Team 1. Suppose Team 0 wins 65% of the time and Team 1 wins the remaining matches. Among the games won by Team 0, only 30% of them come from playing on Team 1 ’s football field. On the other hand, 75% of the victories for Team 1 are obtained while playing at home. If Team 1 is to host the next match between the two teams, which team will most likely emerge as the winner?

let X be the random variable that represents the team hosting the match and Y be the random variable that represents the winner of the match. Both X and Y can take on values from the set $\{0, 1\}$. We can summarize the information given in the problem as follows:

Probability Team 0 wins is $P(Y = 0) = 0.65$.
Probability Team 1 wins is $P(Y = 1) = 1 - P(Y = 0) = 0.35$.
Probability Team 1 hosted the match it won is $P(X = 1|Y = 1) = 0.75$.
Probability Team 1 hosted the match won by Team 0 is $P(X = 1|Y = 0) = 0.3$.

Our objective is to compute $P(Y = 1|X = 1)$, which is the conditional probability that Team 1 wins the next match it will be hosting, and compares it against $P(Y = 0|X = 1)$. Using the Bayes theorem, we obtain

$$\begin{aligned}
P(Y=1|X=1) &= \frac{P(X=1|Y=1)\times P(Y=1)}{P(X=1)} \\
&= \frac{P(X=1|Y=1)\times P(Y=1)}{P(X=1,Y=1)+P(X=1,Y=0)} \\
&= \frac{P(X=1|Y=1)\times P(Y=1)}{P(X=1|Y=1)P(Y=1)+P(X=1|Y=0)P(Y=0)} \\
&= \frac{0.75\times 0.35}{0.75\times 0.35+0.3\times 0.65} \\
&= 0.5738,
\end{aligned}$$

where the law of total probability was applied in the second line. Furthermore, $P(Y=0|X=1) = 1 - P(Y=1|X=1) = 0.4262$. Since $P(Y=1|X=1) > P(Y=0|X=1)$, Team 1 has a better chance than Team 0 of winning the next match.

- **The semantics of Bayesian Network:**
- There are two ways to understand the semantics of the Bayesian network, which is given below:
- **1. To understand the network as the representation of the Joint probability distribution.**
- It is helpful to understand how to construct the network.
- **2. To understand the network as an encoding of a collection of conditional independence statements.**
- It is helpful in designing inference procedure.
- Bayesian networks are graphical models that represent probabilistic relationships among a set of variables. The **semantics** of Bayesian networks revolve around how these relationships are encoded and interpreted. Here are the key aspects of their semantics:

- **Graphical Structure**: A Bayesian network is represented as a directed acyclic graph (DAG), where nodes represent random variables and edges represent direct probabilistic dependencies between variables. The direction of edges indicates the direction of probabilistic influence.
- **Conditional Independence**: The structure of a Bayesian network encodes conditional independence relationships among variables. Two variables are conditionally independent given their parents in the graph. This property allows efficient computation of joint probabilities using the chain rule of probability.
- **Node Parameters**: Each node in a Bayesian network has associated parameters that specify the conditional probability distribution of the node given its parents (conditional probability tables or CPTs). These tables quantify how the probability of a node varies based on the values of its parents.

- **Probabilistic Inference**: Bayesian networks facilitate probabilistic inference, which involves computing probabilities of interest given observed evidence. This is done using algorithms like variable elimination, belief propagation, or sampling methods such as Markov Chain Monte Carlo (MCMC).
- **Causal and Evidential Reasoning**: Bayesian networks can be used for both causal reasoning (understanding how changes in one variable affect others) and evidential reasoning (updating beliefs based on new evidence). The structure of the network helps in distinguishing between these types of reasoning.
- **Learning from Data**: Bayesian networks can be learned from data, where the structure of the graph and the parameters of the nodes are inferred from observed data. This is typically done using algorithms like the Expectation-Maximization (EM) algorithm or various scoring-based approaches.
- **Applications**: Bayesian networks are widely used in various fields such as medicine, finance, genetics, and artificial intelligence for modeling uncertain knowledge, making predictions, diagnosing problems, and decision-making under uncertainty.
- In essence, the semantics of Bayesian networks lie in their ability to represent probabilistic relationships using a graph structure, encode conditional independence assumptions, facilitate probabilistic inference, and support reasoning under uncertainty and learning from data.

Bayesian network

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

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

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

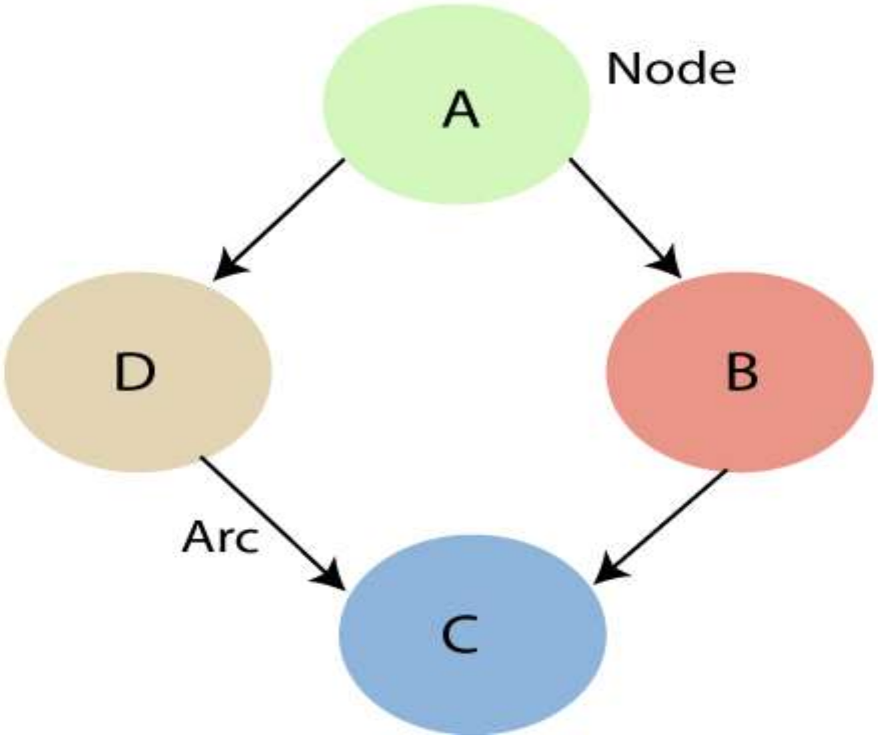
Bayesian Network can be used for building models from data and experts opinions.
it consists of two parts:

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

Bayesian network

The generalized form of Bayesian network that represents and solve decision problems under uncertain knowledge is known as an **Influence diagram**.

Each **node** corresponds to the random variable. **Arc or directed arrows** represent the causal relationship or conditional probabilities between random variable. The Bayesian network graph does not contain any cyclic graph.



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

- The Bayesian network has mainly two components:
- **Causal Component**
- **Actual numbers**
- Each node in the Bayesian network has condition probability distribution **$P(X_i \,|Parent(X_i)\,)$**, which determines the effect of the parent on that node.
- Bayesian network is based on Joint probability distribution and conditional probability.

Bayesian network

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

Problem:

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

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

Bayesian network

P[D, S, A, B, E], can rewrite the above probability statement using joint probability distribution:

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

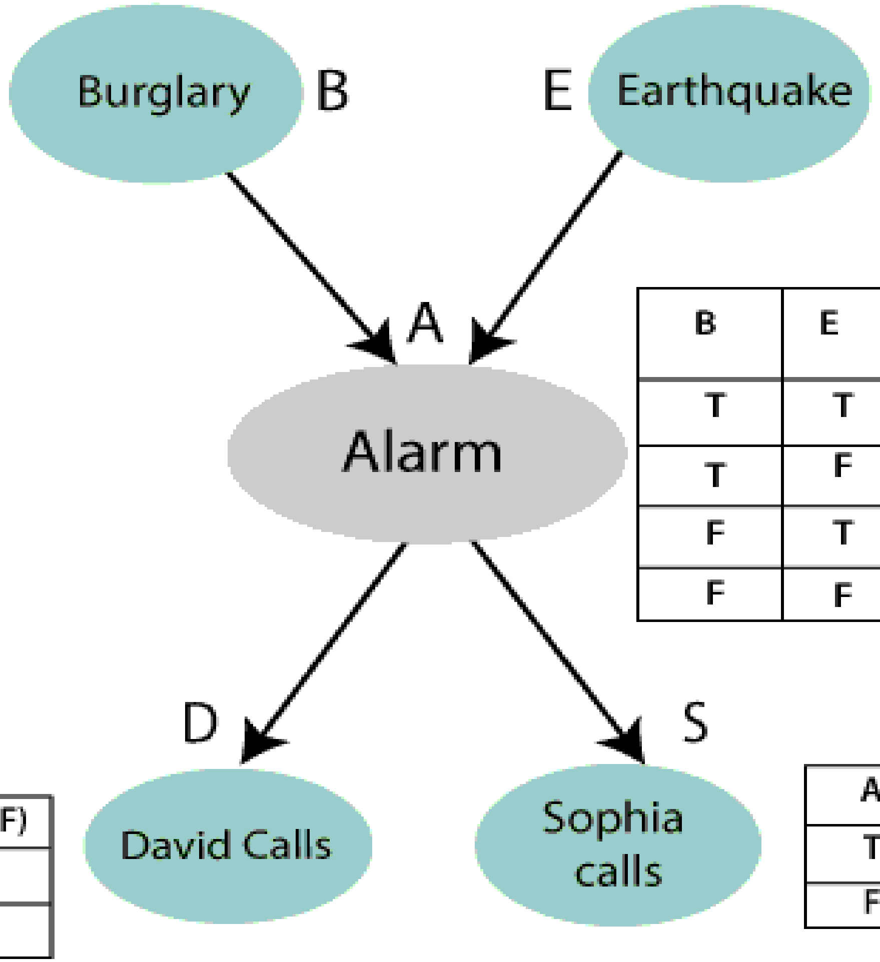
=P[D | S, A, B, E]. P[S | A, B, E]. P[A, B, E]

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

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

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

T	0.002
F	0.998



T	0.001
F	0.999

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

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

Conditional probability table for David Calls:
The Conditional probability of David that he will call depends on the probability of Alarm

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

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

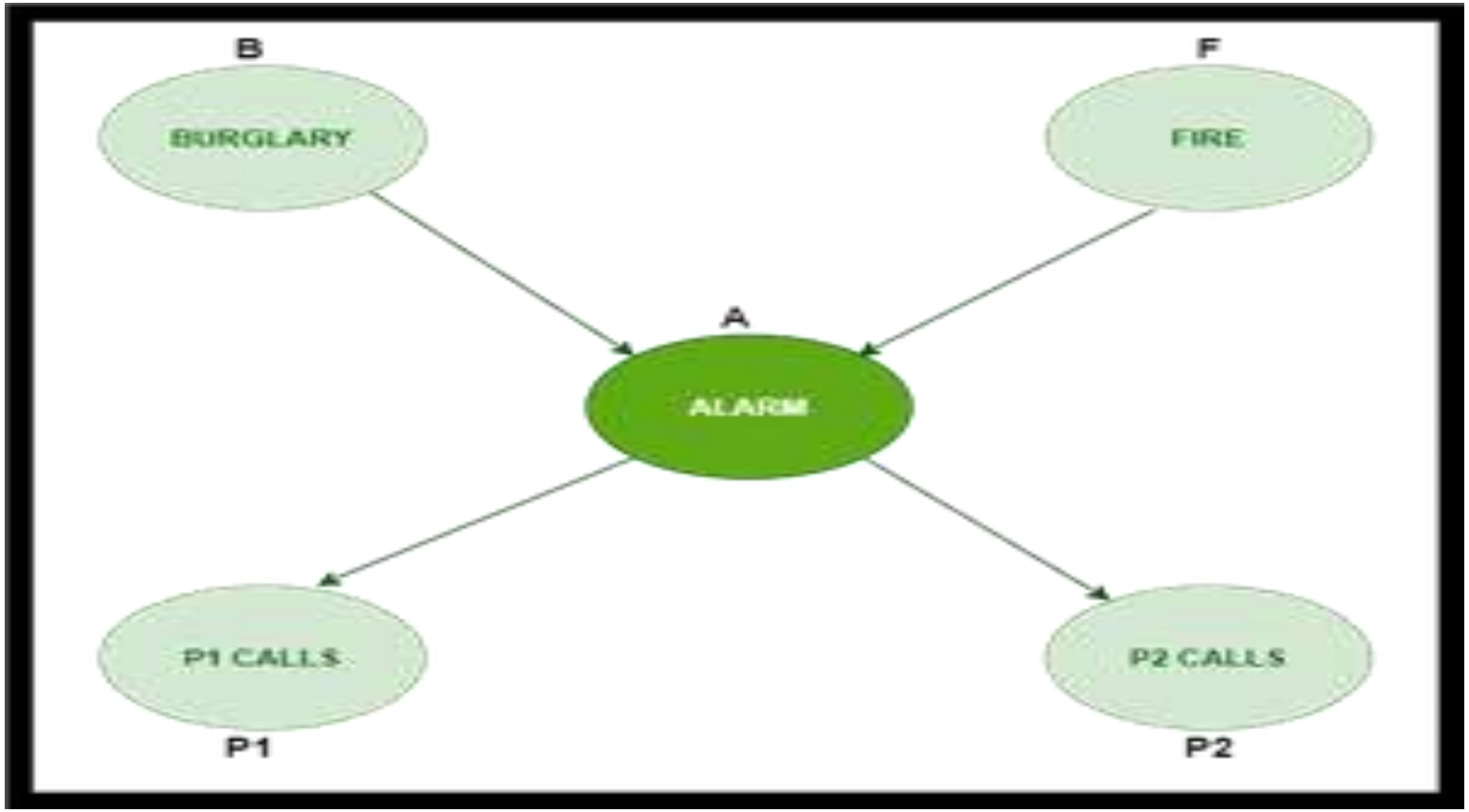
A	P (S=T)	P (S=F)
T	0.75	0.25
F	0.02	0.98

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

- **P(S, D, A, ¬B, ¬E) = P (S|A) *P (D|A)*P (A|¬B ^ ¬E) *P (¬B) *P (¬E).**
- = 0.75* 0.91* 0.001* 0.998*0.999
- **=** 0.00068045.

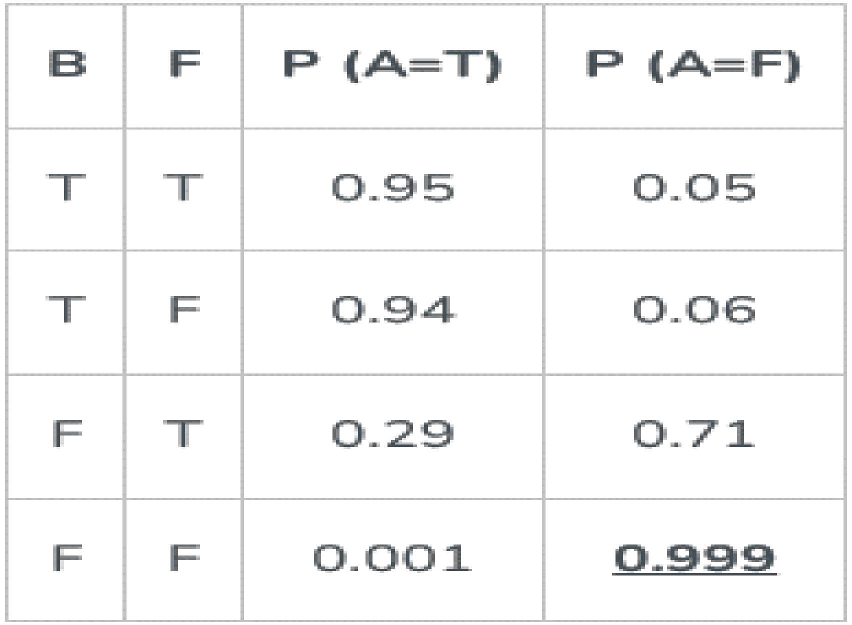
Bayesian network

Bayesian Belief Network is a graphical representation of different probabilistic relationships among random variables in a particular set. It is a classifier with no dependency on attributes i.e it is condition independent. Due to its feature of joint probability, the probability in Bayesian Belief Network is derived, based on a condition — P(attribute/parent) i.e probability of an attribute, true over parent attribute.



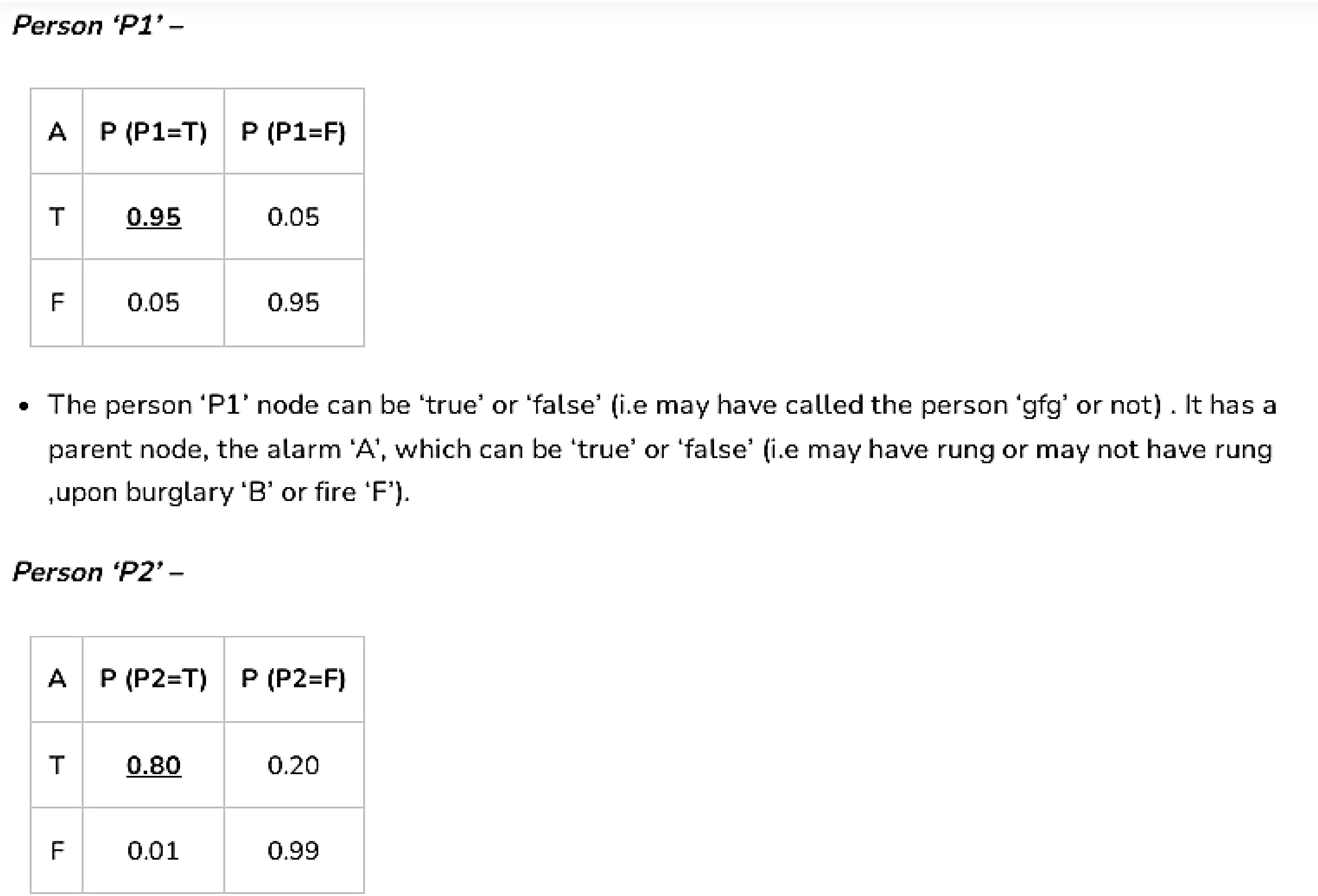
- In the above figure, we have an alarm ‘A’ – a node, say installed in a house of a person ‘gfg’, which rings upon two probabilities i.e burglary ‘B’ and fire ‘F’, which are – parent nodes of the alarm node. The alarm is the parent node of two probabilities P1 calls ‘P1’ & P2 calls ‘P2’ person nodes.
- Upon the instance of burglary and fire, ‘P1’ and ‘P2’ call person ‘gfg’, respectively. But, there are few drawbacks in this case, as sometimes ‘P1’ may forget to call the person ‘gfg’, even after hearing the alarm, as he has a tendency to forget things, quick. Similarly, ‘P2’, sometimes fails to call the person ‘gfg’, as he is only able to hear the alarm, from a certain distance.
- **Q)** Find the probability that ‘P1’ is true (P1 has called ‘gfg’), ‘P2’ is true (P2 has called ‘gfg’) when the alarm ‘A’ rang, but no burglary ‘B’ and fire ‘F’ has occurred.
- => **P (P1, P2, A, ~B, ~F)** [where- P1, P2 & A are ‘true’ events and ‘~B’ & ‘~F’ are ‘false’ events]

- ***Burglary 'B' –***
- **P (B=T) = 0.001** ('B' is true i.e burglary has occurred)
- **P (B=F) = 0.999** ('B' is false i.e burglary has not occurred)
- ***Fire 'F' –***
- **P (F=T) = 0.002** ('F' is true i.e fire has occurred)
- **P (F=F) = 0.998** ('F' is false i.e fire has not occurred)



B	F	P (A=T)	P (A=F)
T	T	0.95	0.05
T	F	0.94	0.06
F	T	0.29	0.71
F	F	0.001	0.999

The alarm 'A' node can be 'true' or 'false' (i.e may have rung or may not have rung). It has two parent nodes burglary 'B' and fire 'F' which can be 'true' or 'false' (i.e may have occurred or may not have occurred) depending upon different conditions.



Person ‘P1’ –

A	P (P1=T)	P (P1=F)
T	**0.95**	0.05
F	0.05	0.95

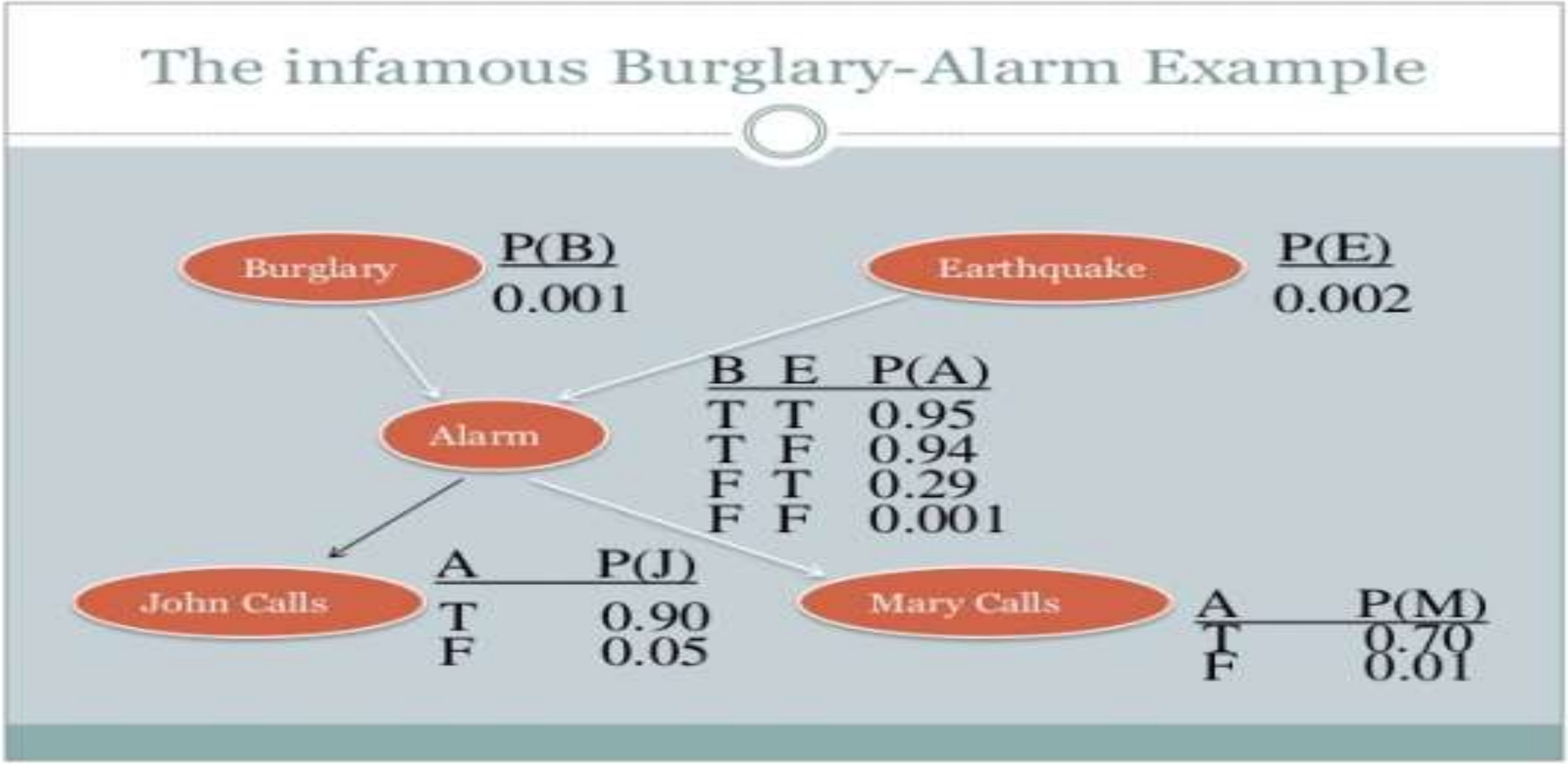
- The person ‘P1’ node can be ‘true’ or ‘false’ (i.e may have called the person ‘gfg’ or not) . It has a parent node, the alarm ‘A’, which can be ‘true’ or ‘false’ (i.e may have rung or may not have rung ,upon burglary ‘B’ or fire ‘F’).

Person ‘P2’ –

A	P (P2=T)	P (P2=F)
T	**0.80**	0.20
F	0.01	0.99

- The person ‘P2’ node can be ‘true’ or false’ (i.e may have called the person ‘gfg’ or not). It has a parent node, the alarm ‘A’, which can be ‘true’ or ‘false’ (i.e may have rung or may not have rung, upon burglary ‘B’ or fire ‘F’).
- **Solution:** Considering the observed probabilistic scan –
- With respect to the question — **P (P1, P2, A, ~B, ~F)** , we need to get the probability of ‘P1’. We find it with regard to its parent node – alarm ‘A’. To get the probability of ‘P2’, we find it with regard to its parent node — alarm ‘A’.
- We find the probability of alarm ‘A’ node with regard to ‘~B’ & ‘~F’ since burglary ‘B’ and fire ‘F’ are parent nodes of alarm ‘A’.
- From the observed probabilistic scan, we can deduce –
- **P (P1, P2, A, ~B, ~F)**
- **= P (P1/A) * P (P2/A) * P (A/~B~F) * P (~B) * P (~F)**
- **= 0.95 * 0.80 * 0.001 * 0.999 * 0.998**
- **= 0.00075**

Bayesian network



The infamous Burglary-Alarm Example

P(B)
0.001

P(E)
0.002

B	E	P(A)
T	T	0.95
T	F	0.94
F	T	0.29
F	F	0.001

A	P(J)
T	0.90
F	0.05

A	P(M)
T	0.70
F	0.01

Bayesian network

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.

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

$= P[x_1 | x_2, x_3, \ldots, x_n] P[x_2, x_3, \ldots, x_n]$

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

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

Bayesian network

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

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

= 0.75* 0.91* 0.001* 0.998*0.999

= 0.00068045.

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

- **Inference in Bayesian** networks refers to the process of using the network to answer probabilistic queries about variables of interest given observed evidence. Here's a structured overview of how inference is conducted in Bayesian networks:
- **1. Types of Queries**
- Inference in Bayesian networks typically involves answering different types of probabilistic queries:
- **Marginal Probability**: Calculate P(X=x), the probability of a specific variable X taking on a particular value x.
- **Conditional Probability**: Calculate P(X=x|Y=y), the probability of X taking on x given that Y is observed to be y.
- **Most Probable Explanation (MPE)**: Determine the most likely assignment of values to a set of variables given evidence, arg maxX P(X|E)) where X are variables of interest and E is the evidence.
- **Parameter Estimation**: Learn or update the parameters (conditional probability tables) of the network given observed data.

- **2. Methods of Inference**
- Several algorithms are used for performing inference in Bayesian networks:
- **Variable Elimination**: A systematic method for computing marginal probabilities and most probable explanations by eliminating variables in a specific order based on their influence.
- **Junction Tree Algorithm (Belief Propagation)**: Efficiently computes marginals in Bayesian networks by transforming the network into a junction tree, which exploits the properties of cliques and separators.
- **Pearl's Message Passing Algorithm**: Specifically designed for polytrees (a type of Bayesian network where each node has at most one parent).
- **Sampling Methods**: Techniques like Markov Chain Monte Carlo (MCMC) methods (e.g., Gibbs sampling) can be used for approximate inference when exact methods are impractical due to computational complexity.
- **Exact Enumeration**: For small networks, exact inference can be performed by explicitly enumerating all possible states of the network.

- **3. Evidence Propagation**
- Before conducting inference, Bayesian networks require evidence to be propagated through the network. Evidence consists of observed values for specific variables. This evidence is used to update probabilities throughout the network, adjusting beliefs based on the observed data.
- **4. Complexity Considerations**
- The computational complexity of inference in Bayesian networks varies based on the structure and size of the network:
- **Graph Structure**: The complexity often depends on the number of variables and the nature of their dependencies (e.g., polytrees are more tractable than general graphs).
- **Exact vs. Approximate**: Exact inference can become computationally expensive for large networks, leading to the use of approximate methods like sampling.

- **5. Applications**
- Inference in Bayesian networks finds applications in various fields:
- **Medical Diagnosis**: Determining the probability of diseases given symptoms.
- **Robotics and AI**: Estimating the location of objects based on sensor data.
- **Finance**: Assessing risk and predicting market trends based on economic indicators.
- In conclusion, inference in Bayesian networks is a fundamental operation that enables reasoning under uncertainty, decision-making based on probabilistic models, and learning from observed data. The choice of inference algorithm depends on the network structure, the type of query, and computational constraints.

- **Exact Inference**
- Inference over a Bayesian network can come in two forms.
- The first is simply evaluating the joint probability of a particular assignment of values for each variable (or a subset) in the network.
- For this, we already have a factorized form of the joint distribution, so we simply evaluate that product using the provided conditional probabilities. If we only care about a subset of variables, we will need to marginalize out the ones we are not interested in. In many cases, this may result in underflow, so it is common to take the logarithm of that product, which is equivalent to adding up the individual logarithms of each term in the product.
- The second, more interesting inference task, is to find P(x|e), or, to find the probability of some assignment of a subset of the variables (x) given assignments of other variables (our evidence, e). In the above example, an example of this could be to find P(Sprinkler, WetGrass | Cloudy), where {Sprinkler, WetGrass} is our x, and {Cloudy} is our e. In order to calculate this, we use the fact that P(x|e) = P(x, e) / P(e) = αP(x, e), where α is a normalization constant that we will calculate at the end such that P(x|e) + P(¬x | e) = 1. In order to calculate P(x, e), we must marginalize the joint probability distribution over the variables that do not appear in x or e, which we will denote as Y.

- **Approximate Inference**
- ***What is approximate inference?***
- It is a method of estimating probabilities in Bayesian networks also called ‘Monte Carlo’ algorithms.
- two types of algorithms: ***Direct sampling*** and ***Markov chain sampling.***
- ***Why use approximate inference?***
- Exact inference becomes intractable for large multiply-connected networks
- Variable elimination can have exponential time and space complexity
- Exact inference is strictly HARDER than NP-complete problems(#P-hard)
- **Direct Sampling**
- In direct sampling , we take samples of events. We expect the frequency of the samples to converge on the probability of the event.

- **Rejection Sampling —**
- Used to compute conditional probabilities P(X|e)
- Generate samples as before
- Reject samples that do not match evidence
- Estimate by counting the how often event X is in the resulting samples
- **Likelihood Weighting —**
- Avoid inefficiency of rejection sampling
- Fix values for evidence variables and only sample the remaining variables
- Weight samples with regard to how likely they are
- **Markov Chain Sampling**
- Generate events by making a random change to the preceding event
- This change is made using the Markov Blanket of the variable to be changed
- Markov Blanket = parents, children, children's parents
- Tally and normalize results

- **Approximate Inference Techniques**
- **Sampling Methods**
- Sampling methods are probabilistic techniques used to approximate complex distributions in Bayesian Networks by generating and analyzing representative samples. Some of the sampling methods are:
- **Monte Carlo Methods**: Monte Carlo methods use random sampling to estimate numerical results. They are particularly useful for high-dimensional integrals and summations in Bayesian Networks. By repeatedly sampling from the probability distributions of interest, these methods provide approximations of desired quantities.
- **Markov Chain Monte Carlo (MCMC)**: MCMC methods generate samples from a probability distribution by constructing a Markov chain that has the desired distribution as its equilibrium distribution. Common MCMC algorithms include the Metropolis-Hastings algorithm and Gibbs sampling. These methods are powerful for exploring complex distributions but can be computationally intensive.
- **Gibbs Sampling**: Gibbs Sampling is a specific type of MCMC method that iteratively samples each variable from its conditional distribution given the current values of all other variables. This technique is effective for high-dimensional spaces and can converge to the target distribution under appropriate conditions.

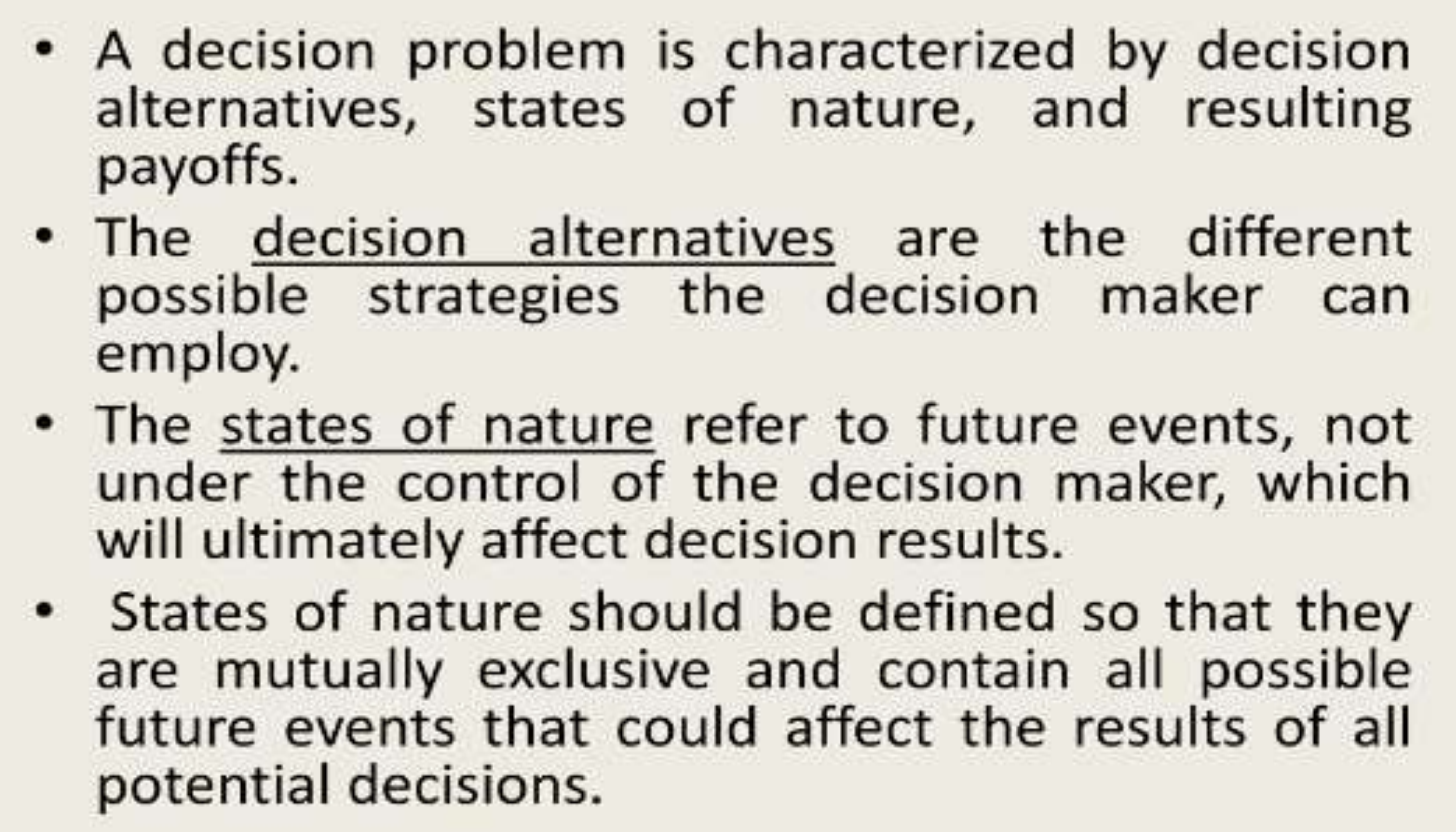
- **Monte Carlo Methods**
- Monte Carlo methods use random sampling to approximate complex mathematical or physical systems. The principle is to generate a large number of random samples from a probability distribution and use these samples to estimate the properties of the distribution.
- This process involves the following steps:
- **Define the Problem**: Identify the quantity to be estimated (e.g., an integral or a probability).
- **Generate Random Samples**: Use a random number generator to produce samples from the distribution of interest.
- **Compute the Estimate**: Calculate the desired quantity using the generated samples, often by averaging the results of the sampled data.
- Estimating the value of π by randomly placing points in a square that encloses a quarter circle and calculating the ratio of points inside the quarter circle to the total number of points.

Decision Theory

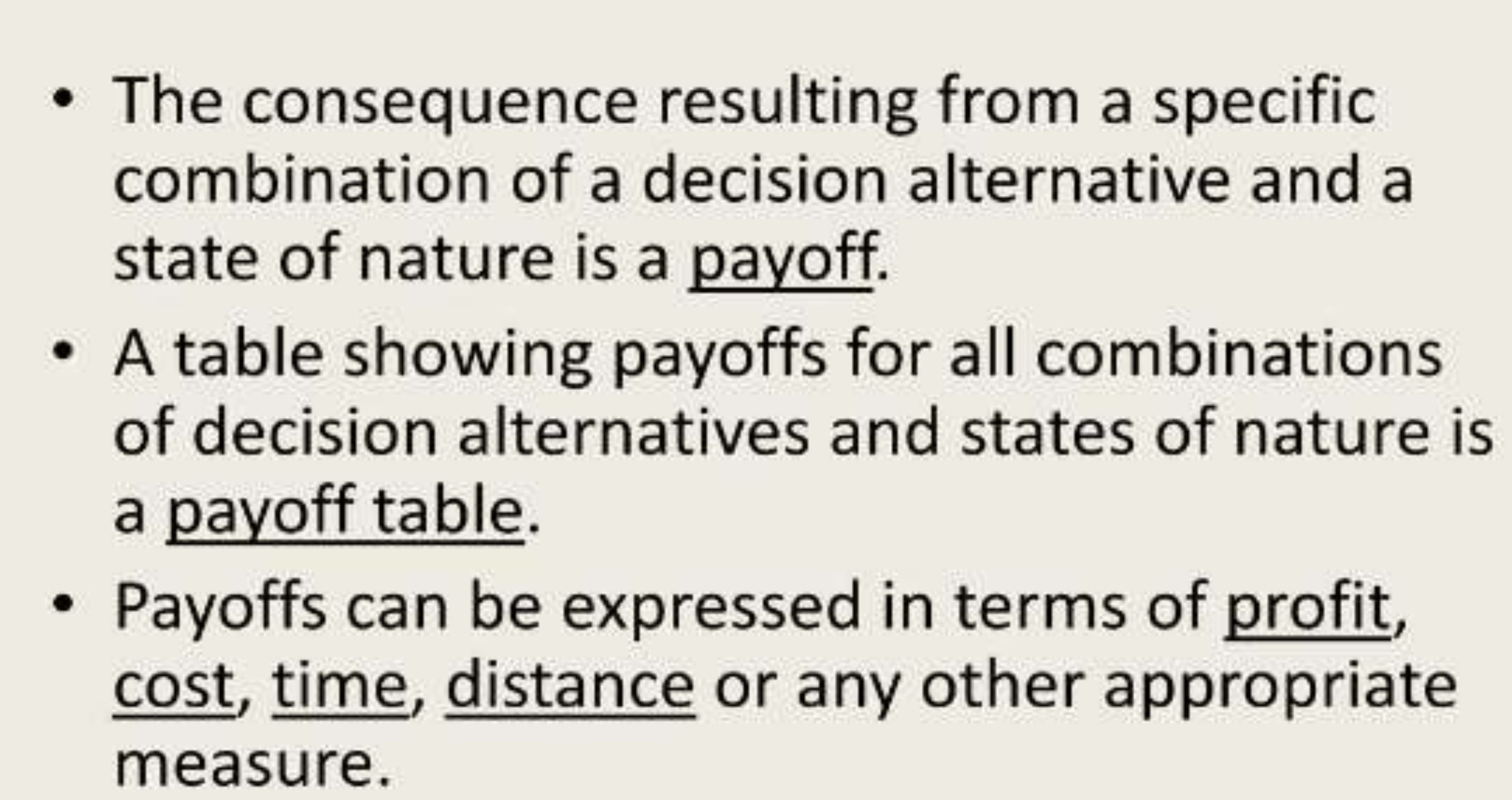
“ **Decision theory** is the study of principles and algorithms for making correct decisions “

Decisions that allow an agent to achieve better outcomes with respect to its goals. Every action at least implicitly represents a decision under uncertainty: in a state of partial knowledge, something has to be done, even if that something turns out to be nothing (call it "the null action"). Even if you don't know how you make decisions, decisions do get made, and so there has to be some underlying mechanism.

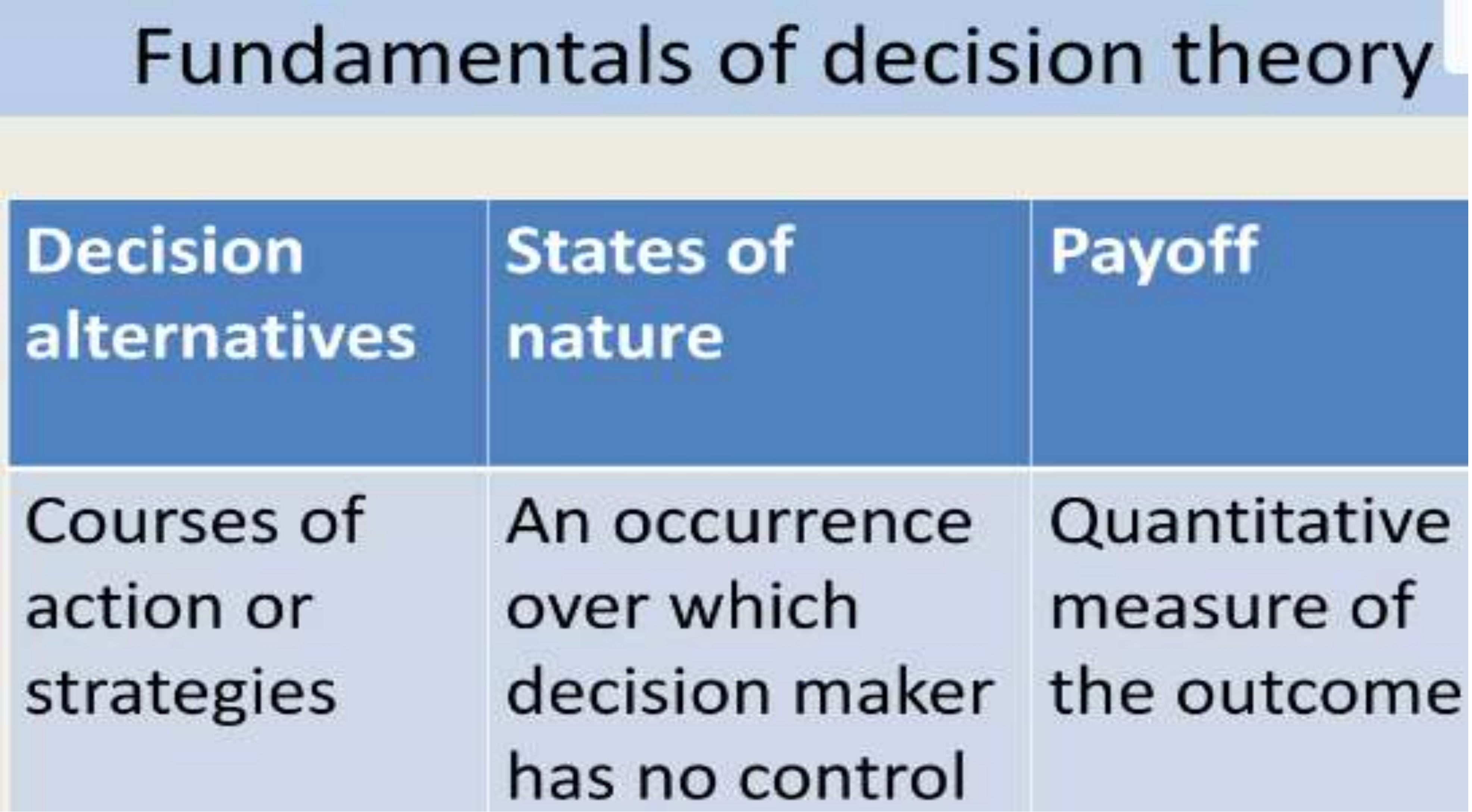
What is it? And how can it be done better? Decision theory has the answers.



- A decision problem is characterized by decision alternatives, states of nature, and resulting payoffs.
- The decision alternatives are the different possible strategies the decision maker can employ.
- The states of nature refer to future events, not under the control of the decision maker, which will ultimately affect decision results.
- States of nature should be defined so that they are mutually exclusive and contain all possible future events that could affect the results of all potential decisions.

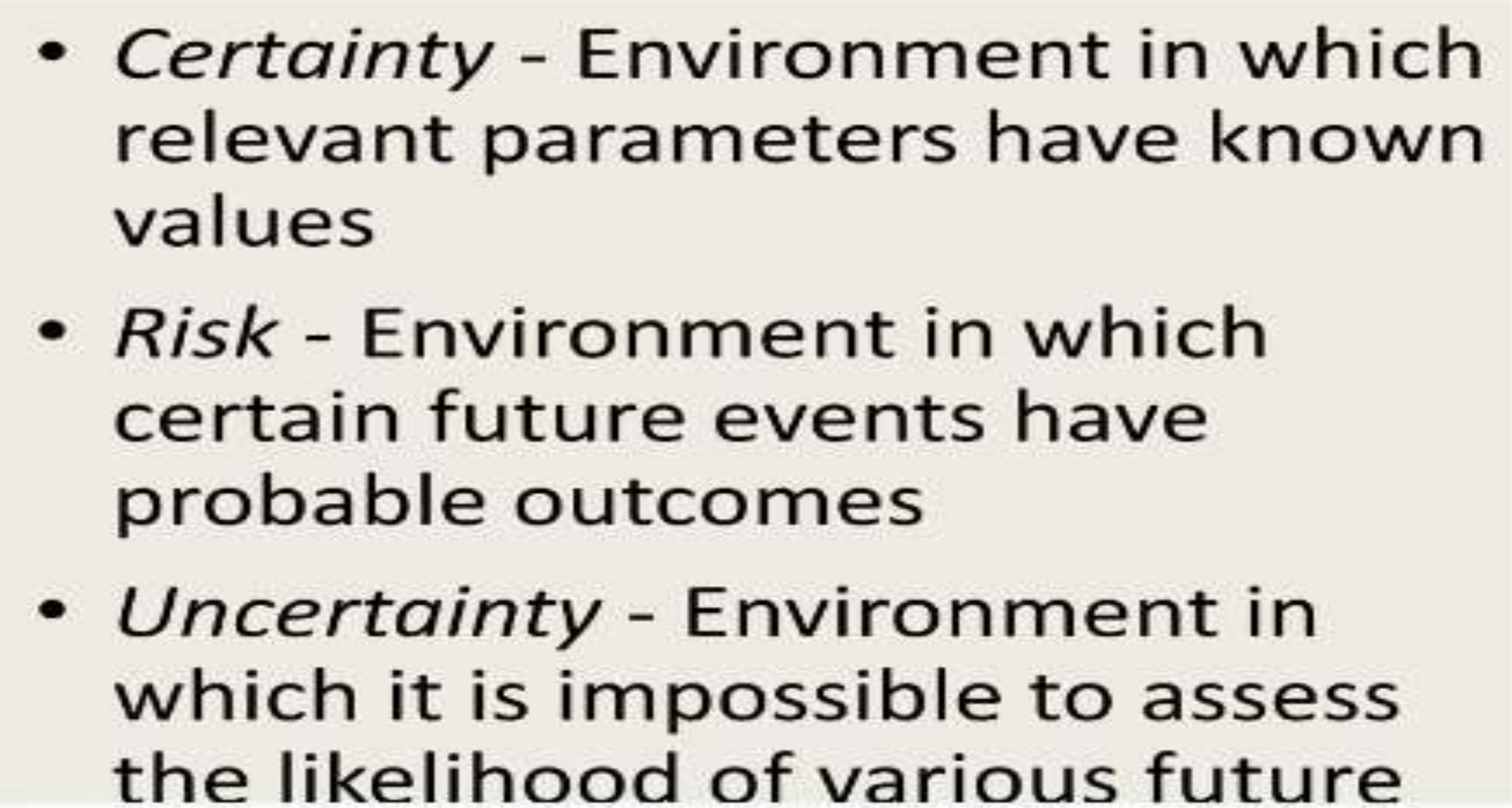


- The consequence resulting from a specific combination of a decision alternative and a state of nature is a payoff.
- A table showing payoffs for all combinations of decision alternatives and states of nature is a payoff table.
- Payoffs can be expressed in terms of profit, cost, time, distance or any other appropriate measure.

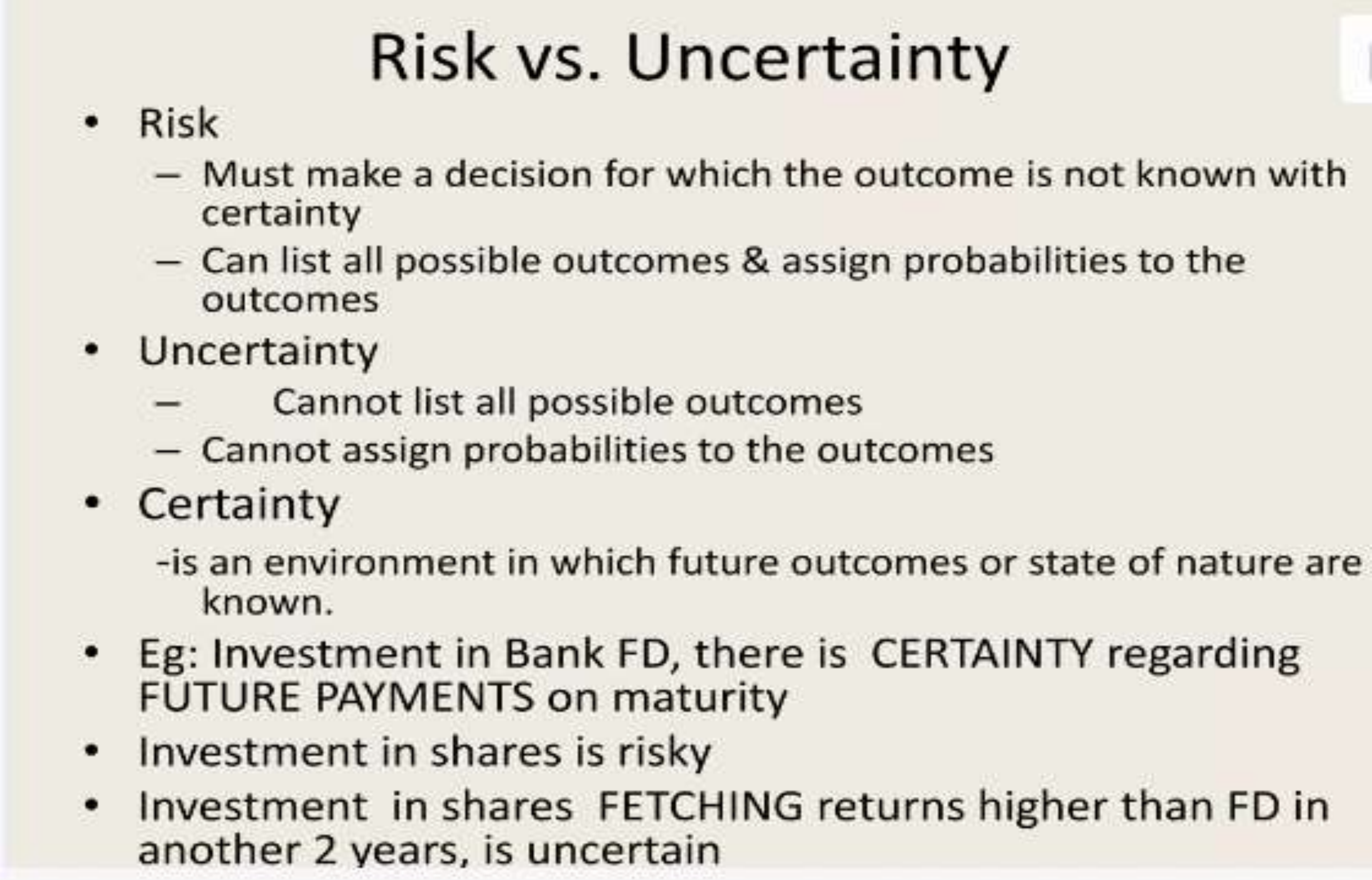


Fundamentals of decision theory

Decision alternatives	States of nature	Payoff
Courses of action or strategies	An occurrence over which decision maker has no control	Quantitative measure of the outcome



- *Certainty* - Environment in which relevant parameters have known values
- *Risk* - Environment in which certain future events have probable outcomes
- *Uncertainty* - Environment in which it is impossible to assess the likelihood of various future



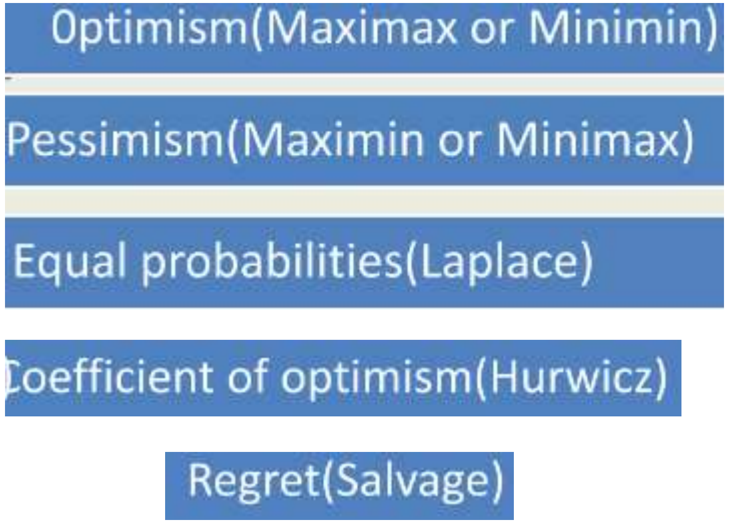
Risk vs. Uncertainty

- Risk
  - Must make a decision for which the outcome is not known with certainty
  - Can list all possible outcomes & assign probabilities to the outcomes
- Uncertainty
  - Cannot list all possible outcomes
  - Cannot assign probabilities to the outcomes
- Certainty

  -is an environment in which future outcomes or state of nature are known.
- Eg: Investment in Bank FD, there is CERTAINTY regarding FUTURE PAYMENTS on maturity
- Investment in shares is risky
- Investment in shares FETCHING returns higher than FD in another 2 years, is uncertain

Criteria of decision making under uncertaininity

Decision-makers must consider multiple possible outcomes and their probabilities in such cases. There are several techniques that decision-makers can use to make decisions under uncertainty, including the Laplace criterion, Maximin, Maximax, Hurwicz, and Minimax regret.



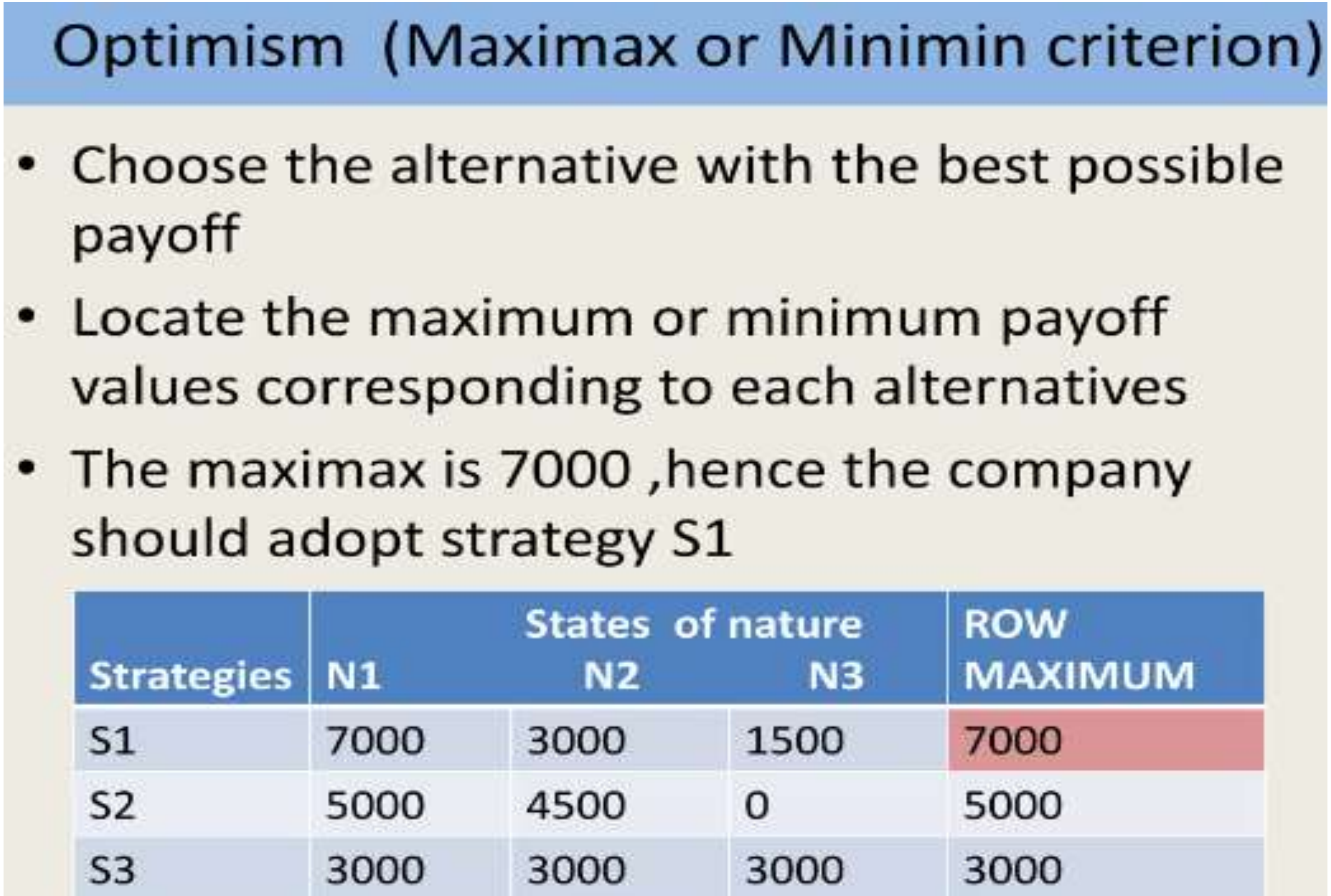
Optimism(Maximax or Minimin)

Pessimism(Maximin or Minimax)

Equal probabilities(Laplace)

Coefficient of optimism(Hurwicz)

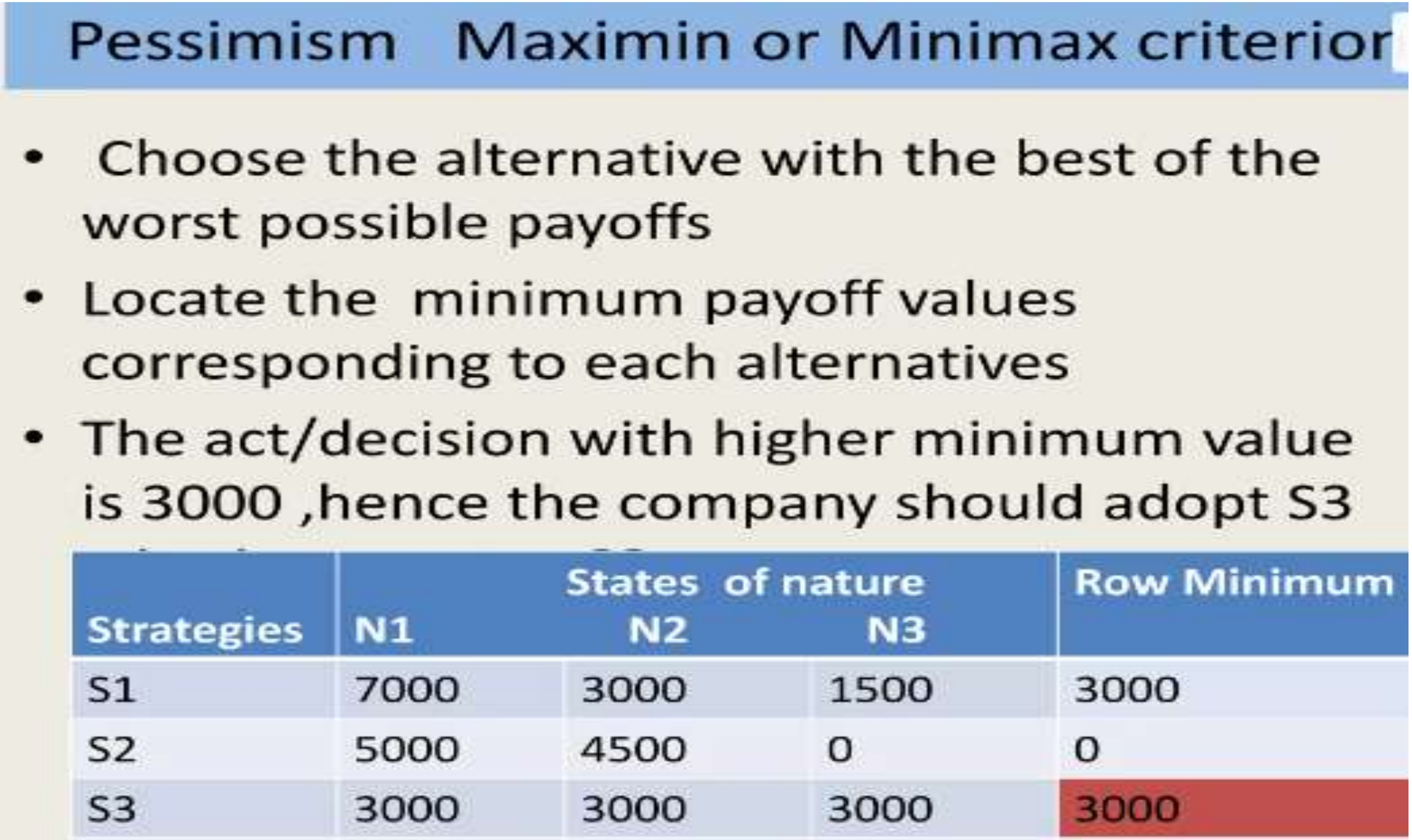
Regret(Salvage)



Optimism (Maximax or Minimin criterion)

- Choose the alternative with the best possible payoff
- Locate the maximum or minimum payoff values corresponding to each alternatives
- The maximax is 7000 ,hence the company should adopt strategy S1

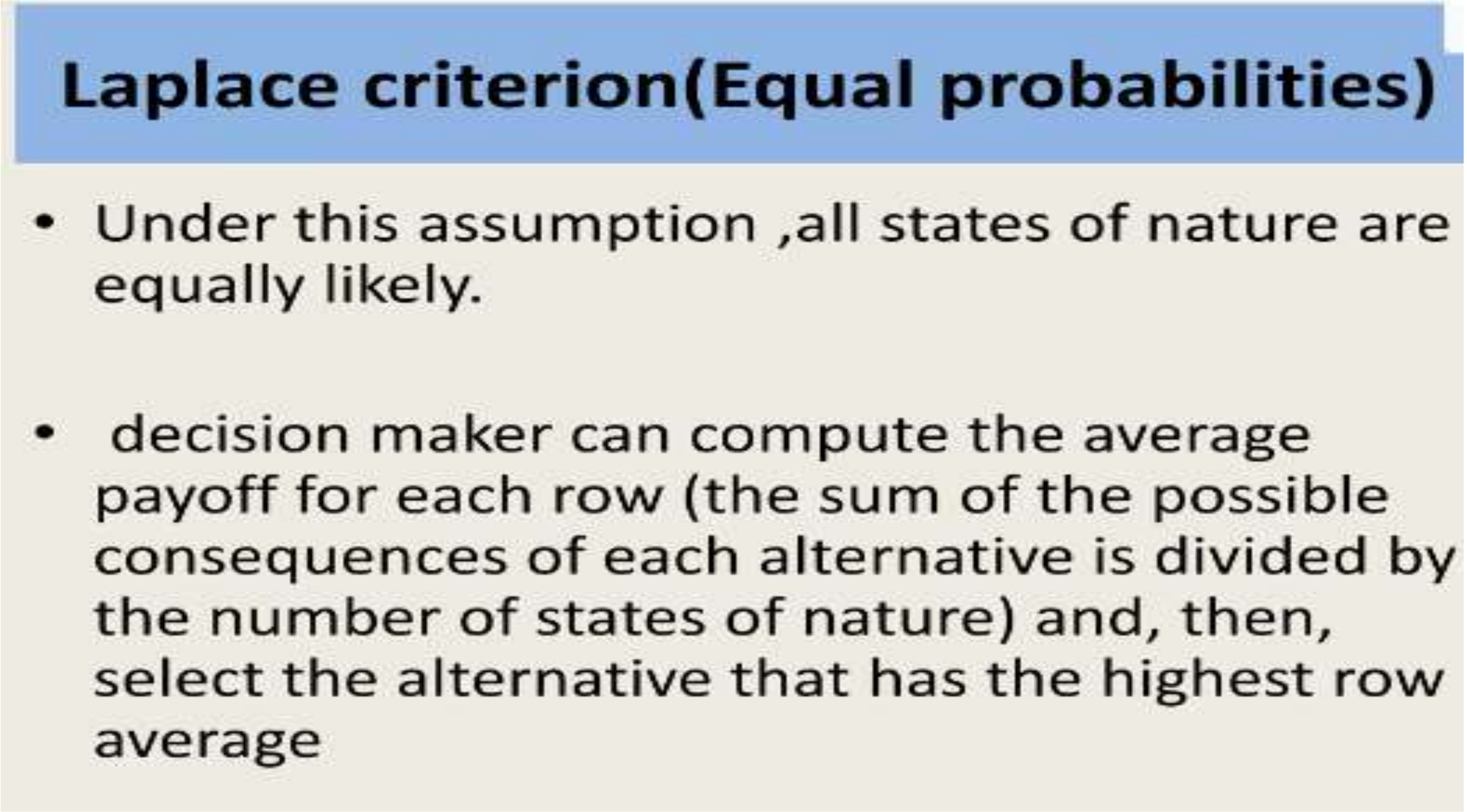
	States of nature			ROW MAXIMUM
Strategies	N1	N2	N3	
S1	7000	3000	1500	7000
S2	5000	4500	0	5000
S3	3000	3000	3000	3000



Pessimism Maximin or Minimax criterior

- Choose the alternative with the best of the worst possible payoffs
- Locate the minimum payoff values corresponding to each alternatives
- The act/decision with higher minimum value is 3000 ,hence the company should adopt S3

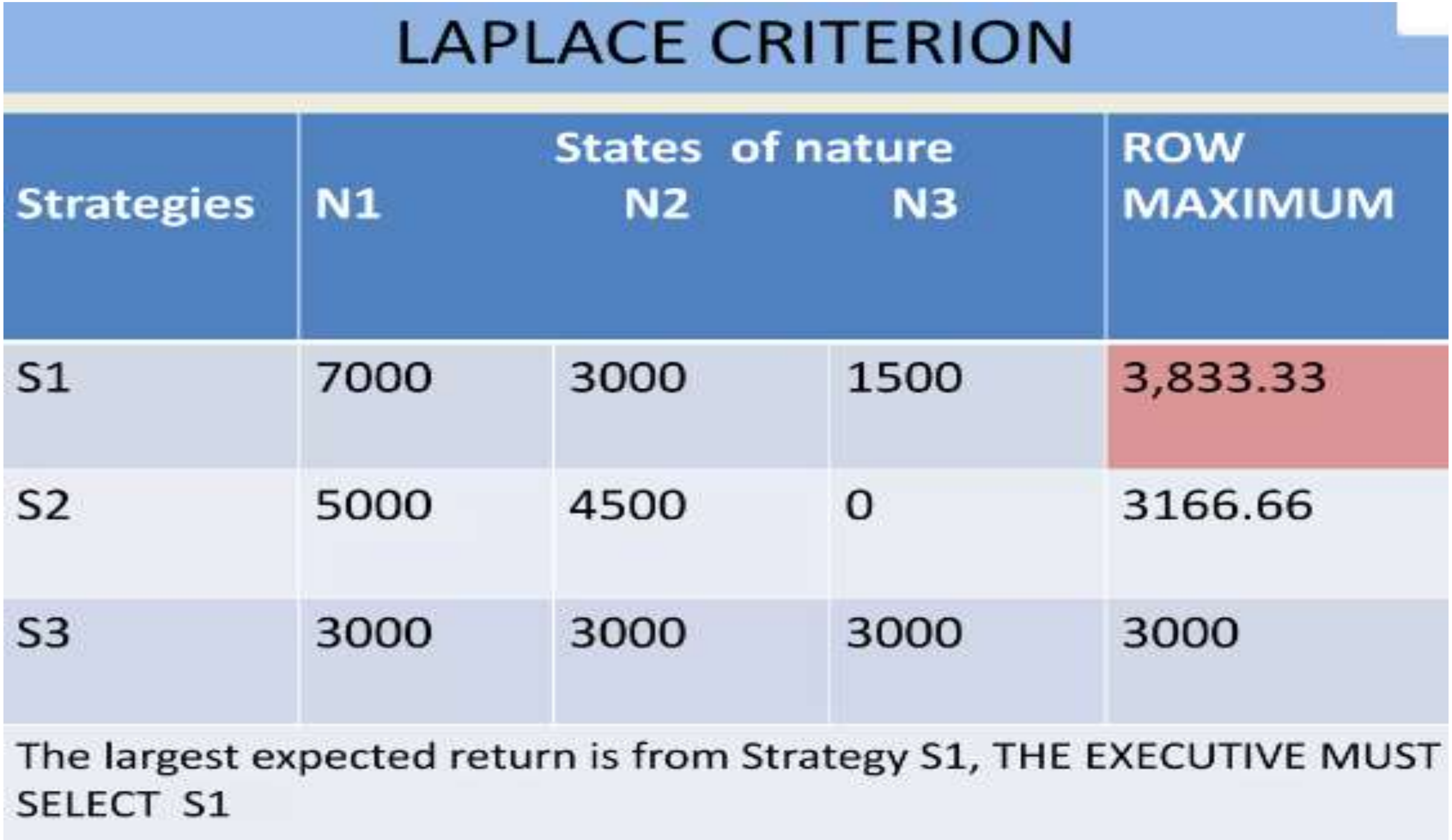
	States of nature			Row Minimum
Strategies	N1	N2	N3	
S1	7000	3000	1500	3000
S2	5000	4500	0	0
S3	3000	3000	3000	3000



Laplace criterion(Equal probabilities)

- Under this assumption ,all states of nature are equally likely.

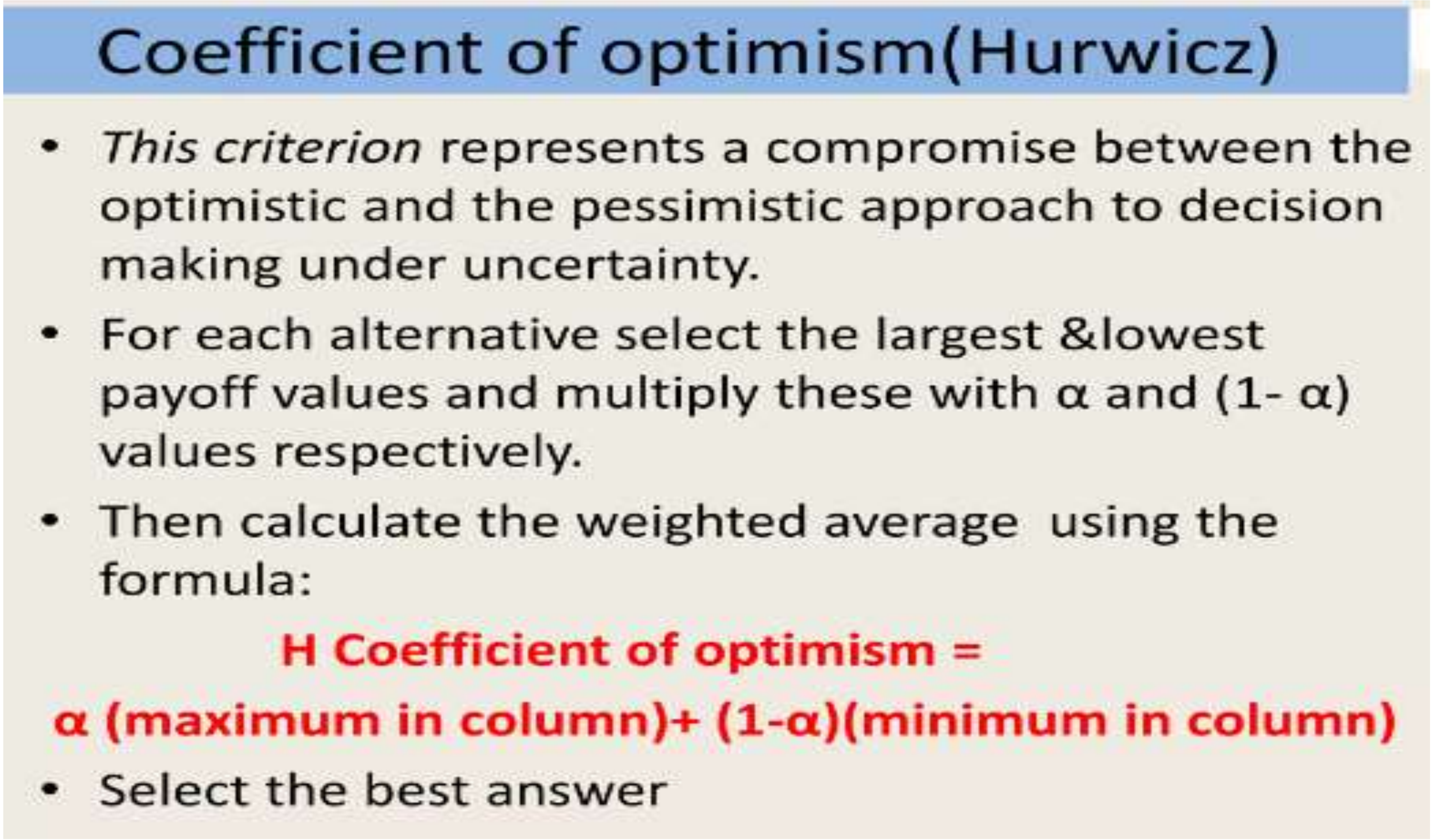
- decision maker can compute the average payoff for each row (the sum of the possible consequences of each alternative is divided by the number of states of nature) and, then, select the alternative that has the highest row average



LAPLACE CRITERION

Strategies	States of nature			ROW MAXIMUM
	N1	N2	N3	
S1	7000	3000	1500	3,833.33
S2	5000	4500	0	3166.66
S3	3000	3000	3000	3000

The largest expected return is from Strategy S1, THE EXECUTIVE MUST SELECT S1



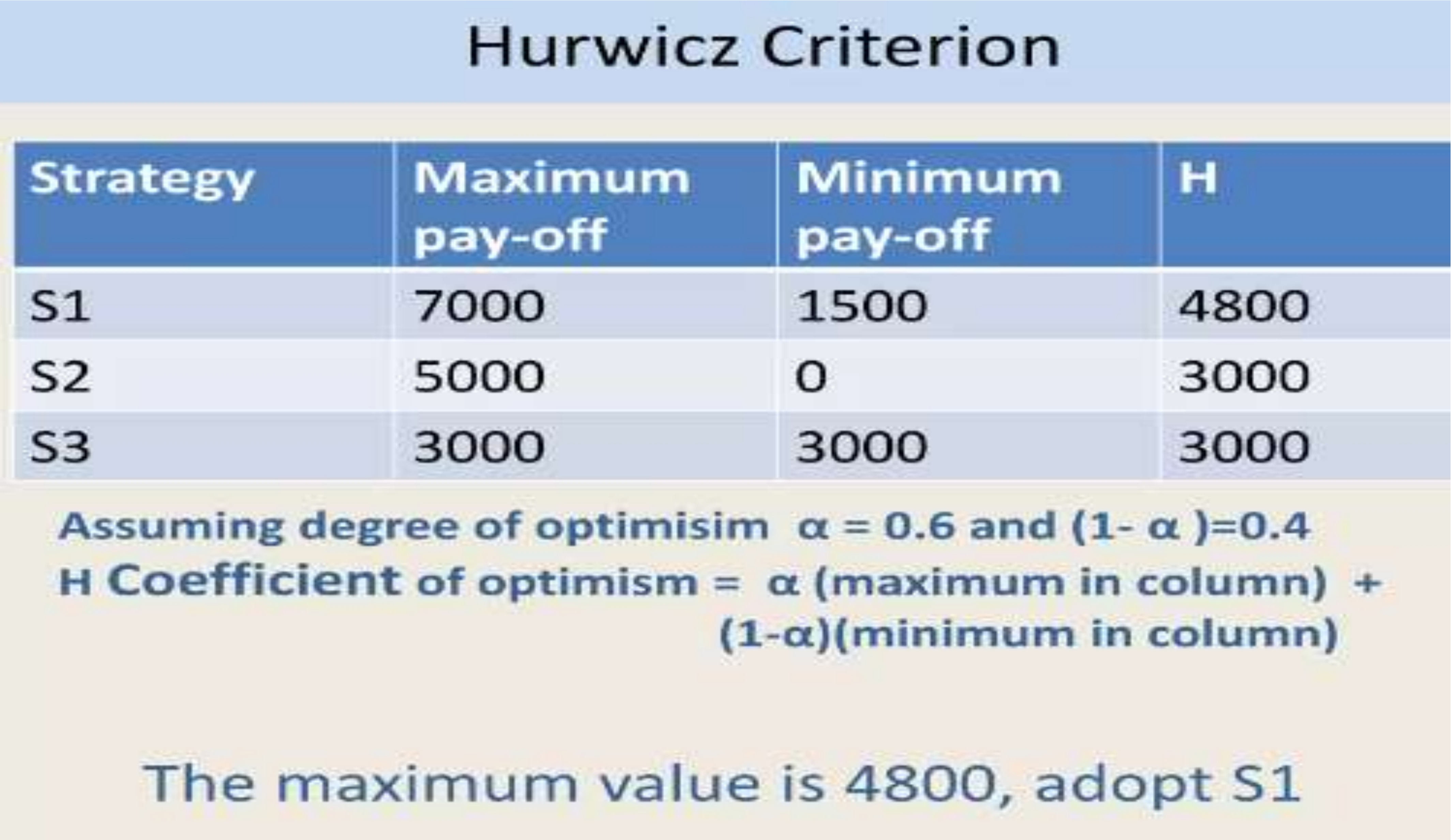
Coefficient of optimism(Hurwicz)

- *This criterion* represents a compromise between the optimistic and the pessimistic approach to decision making under uncertainty.
- For each alternative select the largest &lowest payoff values and multiply these with α and (1- α) values respectively.
- Then calculate the weighted average using the formula:

H Coefficient of optimism =

α (maximum in column)+ (1-α)(minimum in column)

- Select the best answer



Hurwicz Criterion

Strategy	Maximum pay-off	Minimum pay-off	H
S1	7000	1500	4800
S2	5000	0	3000
S3	3000	3000	3000

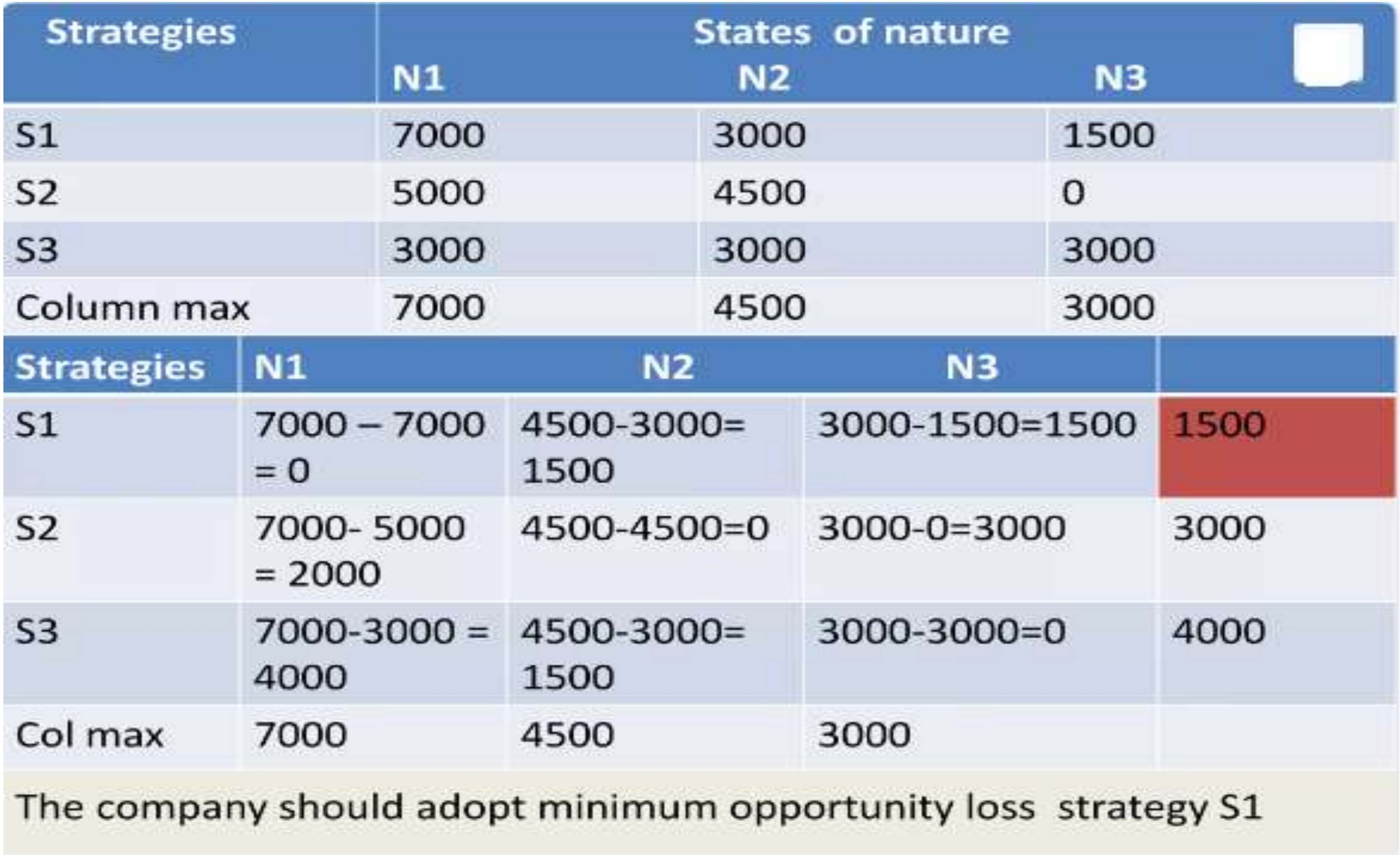
Assuming degree of optimisim α = 0.6 and (1- α)=0.4

H Coefficient of optimism = α (maximum in column) + (1-α)(minimum in column)

The maximum value is 4800, adopt S1

Regret (Salvage rule)

- This rule represents a pessimistic approach.
- The opportunity loss reflects the difference between each payoff and the best possible payoff in a column (it can be defined as the amount of profit foregone by not choosing the best alternative for each state of nature).
- For each course of action identify the maximum regret value, record this no in a row
- Select the course of action with Smallest anticipated opportunity loss value



Strategies	States of nature		
	N1	N2	N3
S1	7000	3000	1500
S2	5000	4500	0
S3	3000	3000	3000
Column max	7000	4500	3000

Strategies	N1	N2	N3	
S1	7000 – 7000 = 0	4500-3000= 1500	3000-1500=1500	1500
S2	7000- 5000 = 2000	4500-4500=0	3000-0=3000	3000
S3	7000-3000 = 4000	4500-3000= 1500	3000-3000=0	4000
Col max	7000	4500	3000	

The company should adopt minimum opportunity loss strategy S1

Markov Decision Process

•A Markov decision process (MDP) refers to a stochastic decision-making process that uses a mathematical framework to model the decision-making of a dynamic system *in scenarios where the results are either random or controlled by a decision maker, which makes sequential decisions over time.*

•MDPs rely on variables such as the environment, agent's actions, and rewards to decide the system's next optimal action. They are classified into four types — finite, infinite, continuous, or discrete — depending on various factors such as sets of actions, available states, and the decision-making frequency.

•In artificial intelligence*,* MDPs model sequential decision-making scenarios with probabilistic dynamics. They are used to design intelligent machines or agents that need to function longer in an environment where actions can yield uncertain results.

- MDP models are typically popular in two sub-areas of AI: probabilistic planning and reinforcement learning(RL).
- Probabilistic planning is the discipline that uses known models to accomplish an agent’s goals and objectives. While doing so, it emphasizes guiding machines or agents to make decisions while enabling them to learn how to behave to achieve their goals.
- Reinforcement learning allows applications to learn from the feedback the agents receive from the environment.

How does a Markov Decision Process work ?

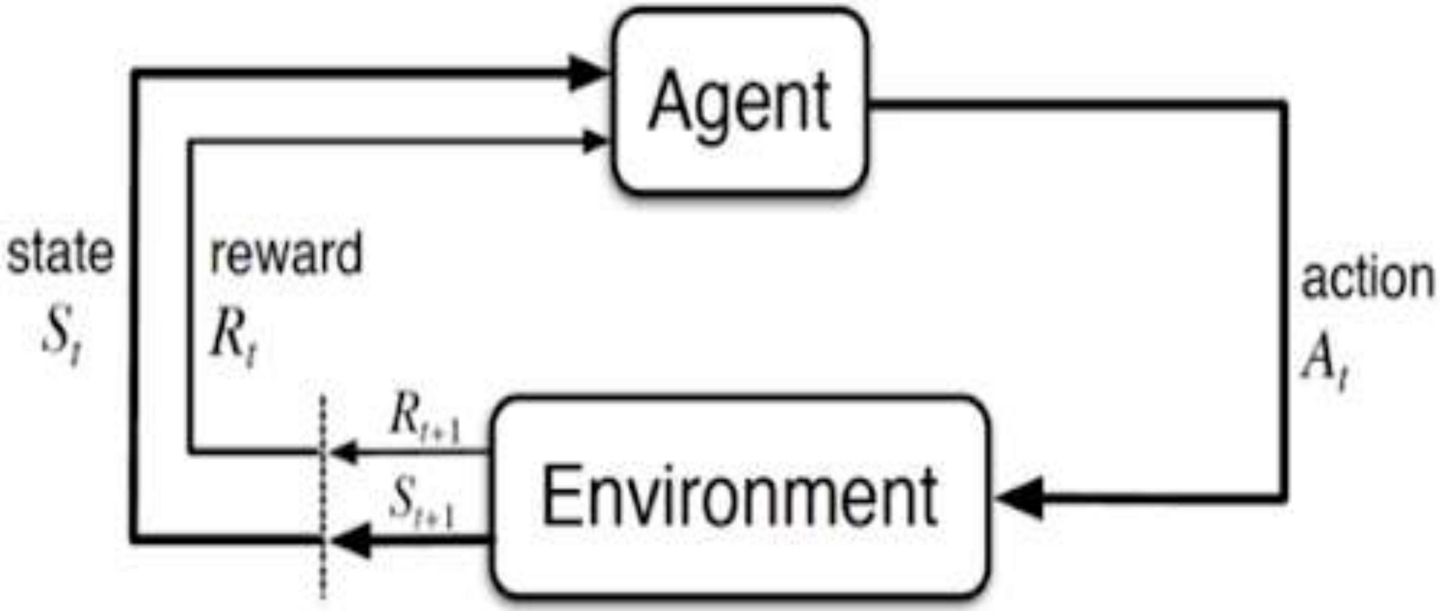
The MDP model operates by using key elements such as the agent, states, actions, rewards, and optimal policies. The agent refers to a system responsible for making decisions and performing actions. It operates in an environment that details the various states that the agent is in while it transitions from one state to another. MDP defines the mechanism of how certain states and an agent's actions lead to the other states. Moreover, the agent receives rewards depending on the action it performs and the state it attains (current state).

- The MDP framework has the following key components:
- S: states ($s \in S$)
- A: Actions $(a \in A)$
- $P\ (St+1|st.at)$: Transition probabilities
- R (s): Reward

- **Agent:** The agent is said to be the learner or a decision-maker who is capable of interacting with their surroundings to achieve a specific goal.
- **Environment:** Everything external to the agent that the agent interacts with is called the environment
- **Action:** Action is the process by which the agent responds to the perceived state of the environment. Based on the state S_t, the agent is capable of choosing an appropriate action At from the set of possible actions $A(S_t)$.
- **Feedback:** Feedback is the information that the agent receives from the environment after acting. The feedback can be of two forms:
  - **New states(S_{t+1}):** After taking an action A_t, the environment transitions to a new state S_{t+1}. This new state reflects the updated situation in the environment as a result of the agent's actions.
  - **Reward(R_{t+1}):** The agent receives the numerical reward R_{t+1}, which provides the measure of immediate effect of the action. The reward earned by the agent can be beneficial or it may negative impact. The agent uses these rewards to adjust its policy to maximize the cumulative rewards over time.

Markov Decision Process

The graphical representation of the MDP model is as follows:



Markov Decision Process

The MDP model uses the Markov Property, which states that the future can be determined only from the present state that encapsulates all the necessary information from the past. The Markov Property can be evaluated by using this equation:

$$\mathbf{P[St+1|St] = P[St+1\ |S1,S2,S3\ldots\ldots St]}$$

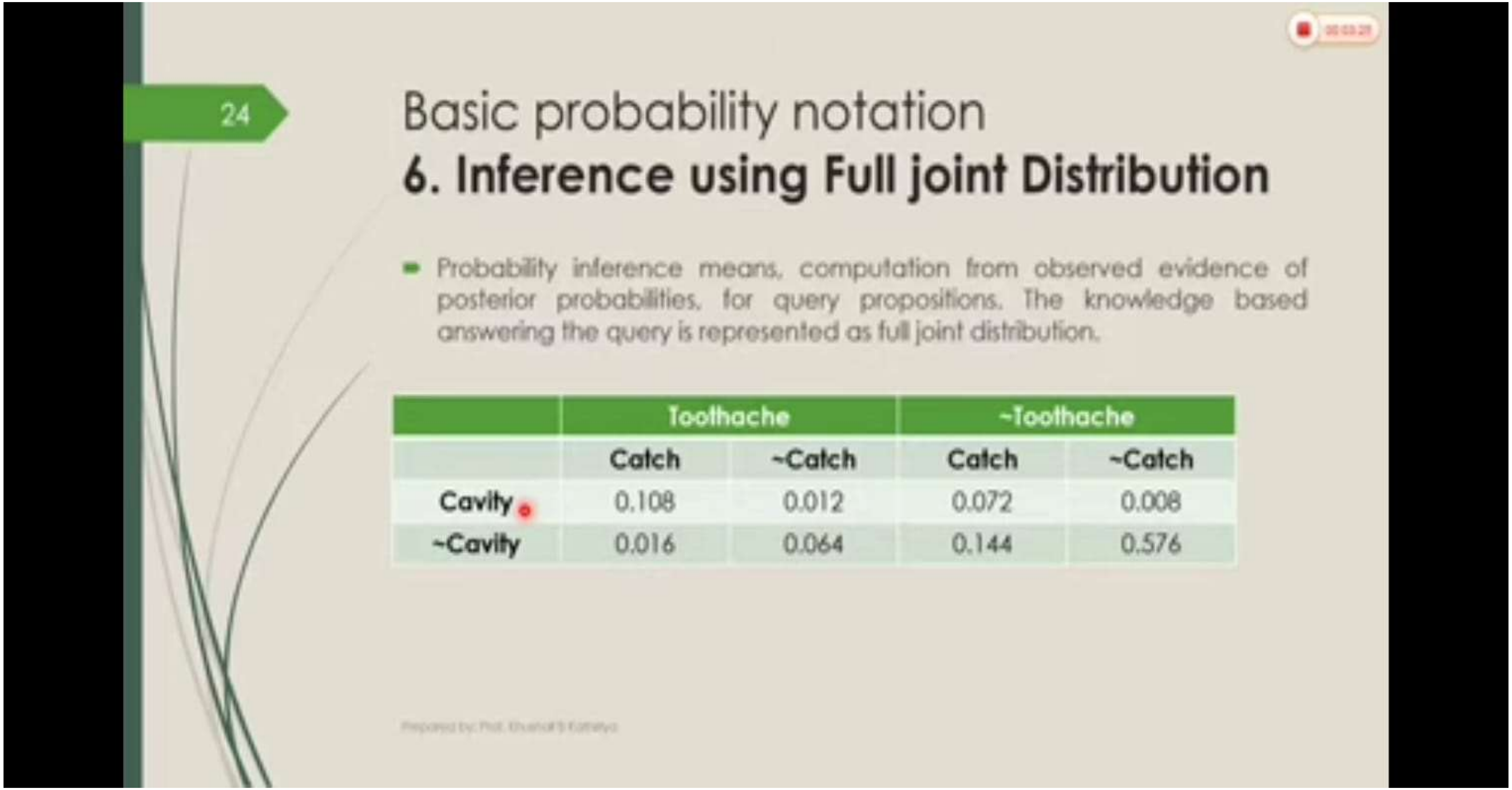
According to this equation, the probability of the next state *(P[St+1])* given the present state *(St)* is given by the next state’s probability *(P[St+1])* considering all the previous states *(S1,S2,S3……St)*.

- Consider a hungry antelope in a wildlife sanctuary looking for food in its environment. It stumbles upon a place with a mushroom on the right and a cauliflower on the left. If the antelope eats the mushroom, it receives water as a reward. However, if it opts for the cauliflower, the nearby lion’s cage opens and sets the lion free in the sanctuary. With time, the antelope learns to choose the side of the mushroom, as this choice offers a valuable reward in return.
- In the above MDP example, two important elements exist — agent and environment. The agent here is the antelope, which acts as a decision-maker. The environment reveals the surrounding (wildlife sanctuary) in which the antelope resides. As the agent performs different actions, different situations emerge. These situations are labeled as states. For example, when the antelope performs an action of eating the mushroom, it receives the reward (water) in correspondence with the action and transitions to another state. The agent (antelope) repeats the process over a period and learns the optimal action at each state.
- In the context of MDP, we can formalize that the antelope knows the optimal action to perform (eat the mushroom). Therefore, it does not prefer eating the cauliflower as it generates a reward that can harm its survival.

Inference using Full joint distribution

The **full joint probability distribution** specifies the probability of each complete assignment of values to random variables. It is usually too large to create or use in its explicit form, but when it is available it can be used to answer queries simply by adding up entries for the possible worlds corresponding to the query propositions.

Probabilistic inference: The computation of posterior probabilities for query propositions given observed evidence.



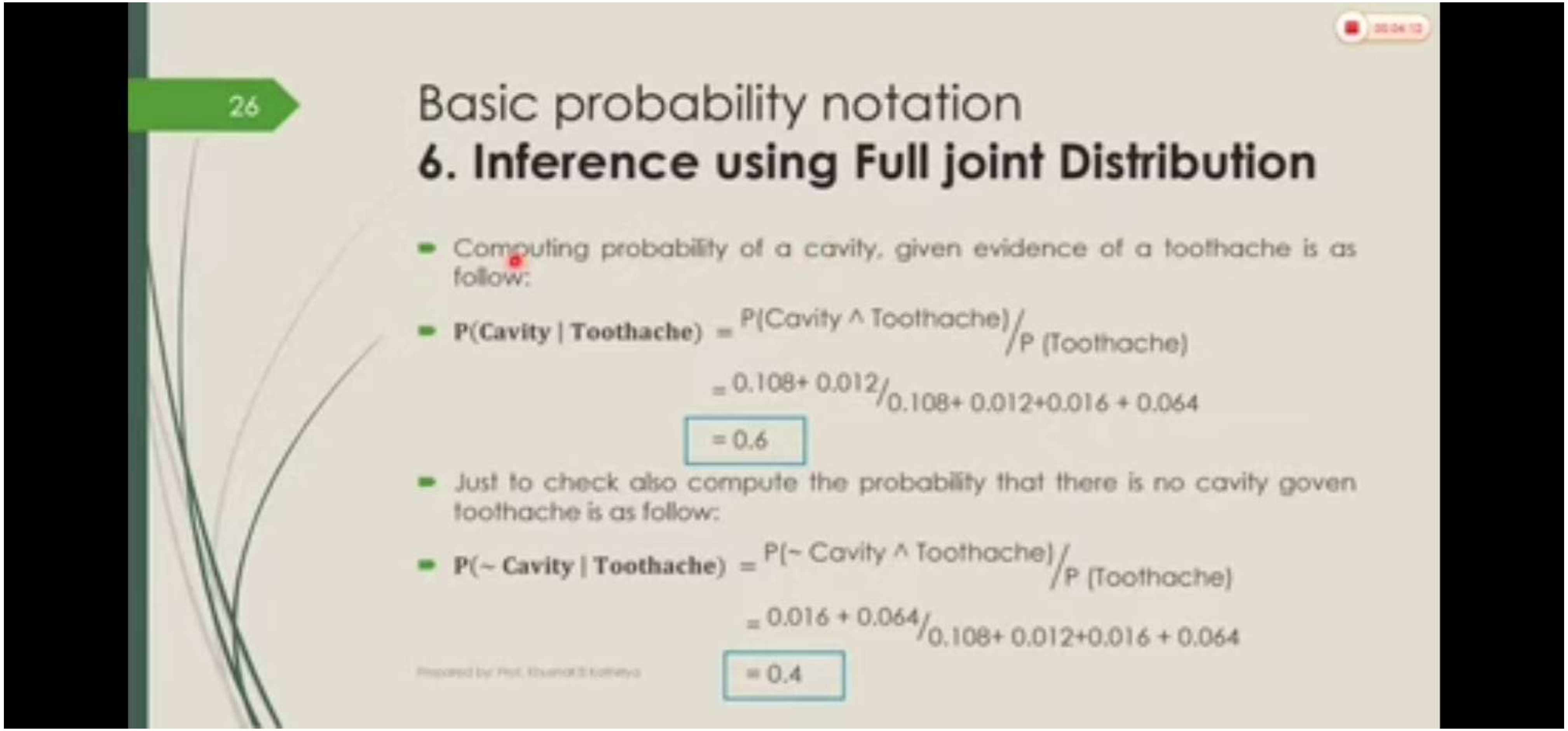
Basic probability notation

6. Inference using Full joint Distribution

- Probability inference means, computation from observed evidence of posterior probabilities, for query propositions. The knowledge based answering the query is represented as full joint distribution.

	Toothache		~Toothache	
	Catch	~Catch	Catch	~Catch
Cavity	0.108	0.012	0.072	0.008
~Cavity	0.016	0.064	0.144	0.576

Inference using Full joint distribution



Basic probability notation

6. Inference using Full joint Distribution

- Computing probability of a cavity, given evidence of a toothache is as follow:
- $\mathbf{P(Cavity \mid Toothache)} = {}^{P(Cavity \wedge Toothache)}/_{P\ (Toothache)}$

$$= {}^{0.108+\ 0.012}/_{0.108+\ 0.012+0.016\ +\ 0.064}$$

$$\boxed{= 0.6}$$

- Just to check also compute the probability that there is no cavity goven toothache is as follow:
- $\mathbf{P(\sim Cavity \mid Toothache)} = {}^{P(\sim Cavity \wedge Toothache)}/_{P\ (Toothache)}$

$$= {}^{0.016\ +\ 0.064}/_{0.108+\ 0.012+0.016\ +\ 0.064}$$

$$\boxed{= 0.4}$$

Hidden Markov Model (HMM)

- HMM is a stochastic model which is built upon the concept of Markov chain based on the assumption that probability of future stats depends only on the current process state rather any state that preceded it.
- For example, when tossing a coin, we cannot say that the result of the fifth toss will be a head.
- This is because a coin does not have any memory and the next result does not depend on the previous result.

- Mathematically, HMM consists of the following variables −
- **States (S)**
- It is a set of hidden or latent states present in a HMM. It is denoted by S.
- **Output symbols (O)**
- It is a set of possible output symbols present in a HMM. It is denoted by O.
- **State Transition Probability Matrix (A)**
- It is the probability of making transition from one state to each of the other states. It is denoted by A.
- **Observation Emission Probability Matrix (B)**
- It is the probability of emitting/observing a symbol at a particular state. It is denoted by B.
- **Prior Probability Matrix (Π)**
- It is the probability of starting at a particular state from various states of the system. It is denoted by Π.

- Hence, a HMM may be defined as $\boldsymbol{\lambda}$ **= (S,O,A,B,$\boldsymbol{\pi}$)**,
- where,
- **S = {s_1,s_2,…,s_N}** is a set of N possible states,
- **O = {o_1,o_2,…,o_M}** is a set of M possible observation symbols,
- A is an **N$\boldsymbol{x}$N** state Transition Probability Matrix (TPM),
- B is an **N$\boldsymbol{x}$M** observation or Emission Probability Matrix (EPM),
- π is an N dimensional initial state probability distribution vector.
- So what makes a Hidden Markov Model. Well suppose you were locked in a room for several days and you were asked about the weather outside. The only piece of evidence you have is whether the person who comes into the room carrying your daily meal is carrying an umbrella or not

	Probability of Umbrella
Sunny	0.1
Rainy	0.8
Foggy	0.3

Table Probabilities of Seeing an Umbrella Based on the Weather

Remember, the equation for the weather Markov process before you were locked in the room was:

$$P(w_1, \ldots, w_n) = \Pi_{i=1}^{n} P(w_i \mid w_{i-1}) \tag{5}$$

Now we have to factor in the fact that the actual weather is *hidden* from you. We do that by using Bayes' Rule:

$$P(w_1, \ldots, w_n \mid u_1, \ldots, u_n) = \frac{P(u_1, \ldots, u_n | w_1, \ldots, w_n) P(w_1, \ldots, w_n)}{P(u_1, \ldots, u_n)} \tag{6}$$

where u_i is true if your caretaker brought an umbrella on day i, and false if the caretaker didn't. The probability $P(w_1, \ldots, w_n)$ is the same as the Markov model from the last section, and the probability $P(u_1, \ldots, u_n)$ is the prior probability of seeing a particular sequence of umbrella events (e.g. {True, False, True}). The probability $P(u_1, \ldots, u_n | w_1, \ldots, w_n)$ can be estimated as $\Pi_{i=1}^{n} P(u_i | w_i)$, if you assume that, for all i, given w_i, u_i is independent of all u_j and w_j, for all $j \neq i$.

$$
\begin{aligned}
\begin{array}{l} P(w_2 = \text{Rainy}| \\ w_1 = \text{Sunny}, u_2 = \text{True}) \end{array} &= \frac{P(w_2 = \text{Rainy}, w_1 = \text{Sunny}|u_2 = \text{T})}{P(w_1 = \text{Sunny}|u_2 = \text{T})} \\
(u_2 \;\; and \;\; w_1 \;\; independent) &= \frac{P(w_2 = \text{Rainy}, w_1 = \text{Sunny}|u_2 = \text{T})}{P(w_1 = \text{Sunny})}
\end{aligned}
$$

$$
\begin{aligned}
(Bayes'\,Rule) &= \frac{P(u_2 = \text{T}|w_1 = \text{Sunny}, w_2 = \text{Rainy})P(w_2 = \text{Rainy}, w_1 = \text{Sunny})}{P(w_1 = \text{Sunny})P(u_2 = \text{T})} \\
(Markov \;\; assumption) &= \frac{P(u_2 = \text{T}|w_2 = \text{Rainy})P(w_2 = \text{Rainy}, w_1 = \text{Sunny})}{P(w_1 = \text{Sunny})P(u_2 = \text{T})} \\
(P(A,B) = P(A|B)P(B)) &= \frac{P(u_2 = \text{T}|w_2 = \text{Rainy})P(w_2 = \text{Rainy}|w_1 = \text{Sunny})P(w_1 = \text{Sunny})}{P(w_1 = \text{Sunny})P(u_2 = \text{T})} \\
(Cancel : P(Sunny)) &= \frac{P(u_2 = \text{T}|w_2 = \text{Rainy})P(w_2 = \text{Rainy}|w_1 = \text{Sunny})}{P(u_2 = \text{T})} \\
&= \frac{(0.8)(0.05)}{0.5} \\
&= .08
\end{aligned}
$$

2. Suppose the day you were locked in the room it was sunny; the caretaker brought in an umbrella on day 2, but not on day 3. Again assuming that the prior probability of the caretaker bringing an umbrella is 0.5, what's the probability that it's foggy on day 3?

$$
\begin{aligned}
\begin{array}{c} P(w_3 = \mathrm{F} \mid \\ w_1 = \mathrm{S},\, u_2 = \mathrm{T},\, u_3 = \mathrm{F}) \end{array}
&= P(w_2 = \mathrm{Foggy},\, w_3 = \mathrm{Foggy} \mid \\
&\qquad\qquad w_1 = \mathrm{Sunny},\, u_2 = \mathrm{True},\, u_3 = \mathrm{False}) + \\
&\quad P(w_2 = \mathrm{Rainy},\, w_3 = \mathrm{Foggy} \mid \ldots) + \\
&\quad P(w_2 = \mathrm{Sunny},\, w_3 = \mathrm{Foggy} \mid \ldots) \\
&= \frac{P(u_3 = F|w_3 = F)P(u_2 = T|w_2 = F)P(w_3 = F|w_2 = F)P(w_2 = F|w_1 = S)P(w_1 = S)}{P(u_3 = F)P(u_2 = T)P(w_1 = S)} + \\
&\quad \frac{P(u_3 = F|w_3 = F)P(u_2 = T|w_2 = R)P(w_3 = F|w_2 = R)P(w_2 = R|w_1 = S)P(w_1 = S)}{P(u_3 = F)P(u_2 = T)P(w_1 = S)} + \\
&\quad \frac{P(u_3 = F|w_3 = F)P(u_2 = T|w_2 = S)P(w_3 = F|w_2 = S)P(w_2 = S|w_1 = S)P(w_1 = S)}{P(u_3 = F)P(u_2 = T)P(w_1 = S)} \\
&= \frac{P(u_3 = F|w_3 = F)P(u_2 = T|w_2 = F)P(w_3 = F|w_2 = F)P(w_2 = F|w_1 = S)}{P(u_3 = F)P(u_2 = T)} + \\
&\quad \frac{P(u_3 = F|w_3 = F)P(u_2 = T|w_2 = R)P(w_3 = F|w_2 = R)P(w_2 = R|w_1 = S)}{P(u_3 = F)P(u_2 = T)} + \\
&\quad \frac{P(u_3 = F|w_3 = F)P(u_2 = T|w_2 = S)P(w_3 = F|w_2 = S)P(w_2 = S|w_1 = S)}{P(u_3 = F)P(u_2 = T)} \\
&= \frac{(0.7)(0.3)(0.5)(0.15)}{(0.5)(0.5)} + \\
&\quad \frac{(0.7)(0.8)(0.2)(0.05)}{(0.5)(0.5)} + \\
&\quad \frac{(0.7)(0.1)(0.15)(0.8)}{(0.5)(0.5)} \\
&= 0.119
\end{aligned}
$$

- Suppose there are a set of data points that need to be grouped into several parts or clusters based on their similarity. In Machine Learning, this is known as Clustering. There are several methods available for clustering:
- K Means Clustering
- Hierarchical Clustering
- Gaussian Mixture Models
- In this article, Gaussian Mixture Model will be discussed.

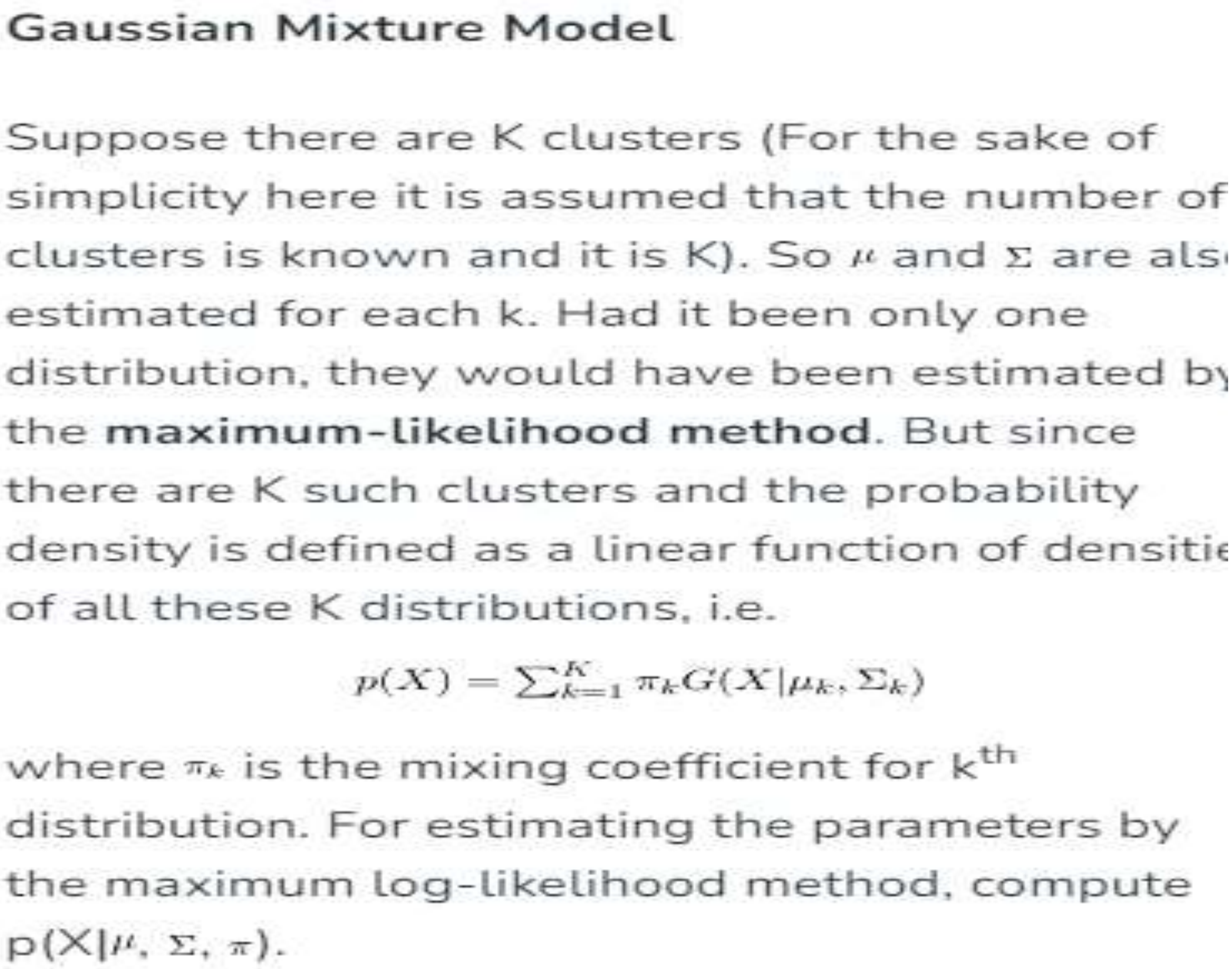
- **Normal or Gaussian Distribution**
- In real life, many datasets can be modeled by Gaussian Distribution (Univariate or Multivariate). So it is quite natural and intuitive to assume that the clusters come from different Gaussian Distributions. Or in other words, it tried to model the dataset as a mixture of several Gaussian Distributions. This is the core idea of this model.
  In one dimension the probability density function of a Gaussian Distribution is given by

$$G(X|\mu,\sigma) = \frac{1}{\sigma\sqrt{2\pi}}e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

where μ and σ^2 are respectively the mean and variance of the distribution. For Multivariate (let us say d-variate) Gaussian Distribution, the probability density function is given by

$$G(X|\mu, \Sigma) = \frac{1}{\sqrt{(2\pi)|\boldsymbol{\Sigma}|}} \exp\left(-\frac{1}{2}(X-\mu)^T\boldsymbol{\Sigma}^{-1}(X-\mu)\right)$$

Here μ is a d dimensional vector denoting the mean of the distribution and Σ is the d X d covariance matrix.

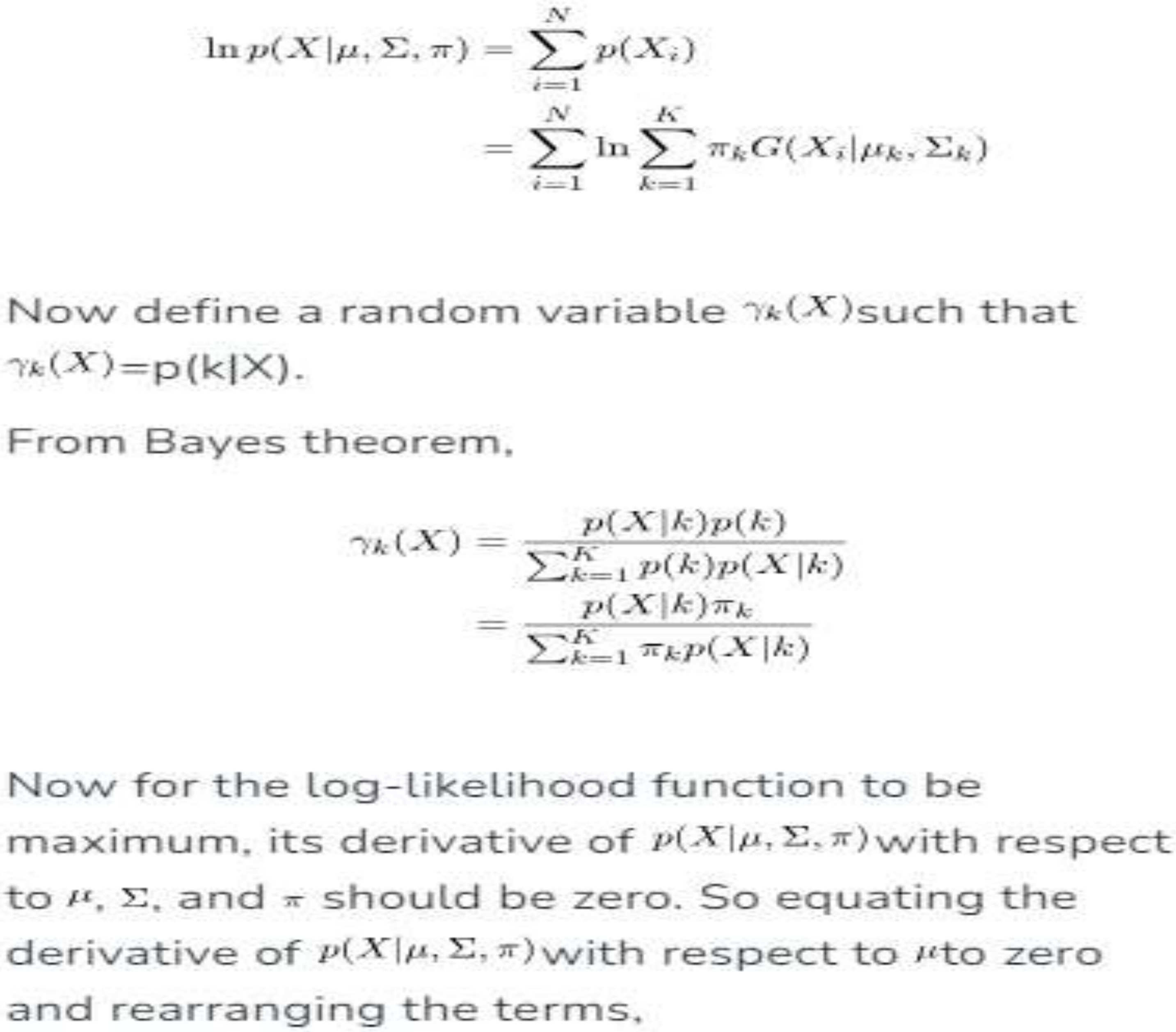


Gaussian Mixture Model

Suppose there are K clusters (For the sake of simplicity here it is assumed that the number of clusters is known and it is K). So μ and Σ are als estimated for each k. Had it been only one distribution, they would have been estimated by the **maximum-likelihood method**. But since there are K such clusters and the probability density is defined as a linear function of densiti of all these K distributions, i.e.

$$p(X) = \sum_{k=1}^{K} \pi_k G(X|\mu_k, \Sigma_k)$$

where π_k is the mixing coefficient for kth distribution. For estimating the parameters by the maximum log-likelihood method, compute p(X|μ, Σ, π).



$$
\begin{aligned}
\ln p(X|\mu,\Sigma,\pi) &= \sum_{i=1}^{N} p(X_i) \\
&= \sum_{i=1}^{N} \ln \sum_{k=1}^{K} \pi_k G(X_i|\mu_k,\Sigma_k)
\end{aligned}
$$

Now define a random variable $\gamma_k(X)$such that $\gamma_k(X)$=p(k|X).

From Bayes theorem,

$$
\begin{aligned}
\gamma_k(X) &= \frac{p(X|k)p(k)}{\sum_{k=1}^{K} p(k)p(X|k)} \\
&= \frac{p(X|k)\pi_k}{\sum_{k=1}^{K} \pi_k p(X|k)}
\end{aligned}
$$

Now for the log-likelihood function to be maximum, its derivative of $p(X|\mu,\Sigma,\pi)$with respect to μ, Σ, and π should be zero. So equating the derivative of $p(X|\mu,\Sigma,\pi)$with respect to $\mu$to zero and rearranging the terms,

$$\mu_k = \frac{\sum_{n=1}^{N} \gamma_k(x_n) x_n}{\sum_{n=1}^{N} \gamma_k(x_n)}$$

Similarly taking the derivative with respect to Σ and pi respectively, one can obtain the following expressions.

$$\Sigma_k = \frac{\sum_{n=1}^{N} \gamma_k(x_n)(x_n - \mu_k)(x_n - \mu_k)^T}{\sum n=1^N \gamma_k(x_n)}$$

And

$$\pi_k = \frac{1}{N} \sum_{n=1}^{N} \gamma_k(x_n)$$

> ***Note:*** $\sum_{n=1}^{N} \gamma_k(x_n)$ *denotes the total number of sample points in the k^{th} cluster. Here it is assumed that there is a total N number of samples and each sample containing d features is denoted by x_i.*