CS 3491 - ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING
UNIT II PROBABILISTIC REASONING

SYLLABUS:

Acting under uncertainty – Bayesian inference – Naïve Bayes models. Probabilistic reasoning – Bayesian networks – exact inference in BN – approximate inference in BN – causal networks.

PART A

1. ***Define uncertainty and list the causes of uncertainty.***

Uncertainty:

- The knowledge representation, A→B, means if A is true then B is true, but a situation where not sure about whether A is true or not then cannot express this statement, this situation is called uncertainty.
- So to represent uncertain knowledge, uncertain reasoning or probabilistic reasoning is used.

Causes of uncertainty:

1. Causes of uncertainty in the real world
2. Information occurred from unreliable sources.
3. Experimental Errors
4. Equipment fault
5. Temperature variation
6. Climate change.

2. ***Define Probabilistic reasoning. Mention the need of probabilistic reasoning in AI***

Probabilistic reasoning:

- Probabilistic reasoning is a way of knowledge representation, the concept of probability is applied to indicate the uncertainty in knowledge.

Need of probabilistic reasoning in AI:

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

3. List the Ways to solve problems with uncertain knowledge.

- Bayes' rule
- Bayesian Statistics

4. Define Probability and the probability of occurrence.

- Probability can be defined as a chance that an uncertain event will occur.
- The value of probability always remains between 0 and 1 that represent ideal uncertainties.
  - $0 \leq P(A) \leq 1$, where P(A) is the probability of an event A.
  - $P(A) = 0$, indicates total uncertainty in an event A.
  - $P(A) =1$, indicates total certainty in an event A.
- Formula to find the probability of an uncertain event

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

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

(¬A) + P(A) = 1.

5. Define the terms event, sample space, random variables, prior probability and posterior probability.

- **Event:** Each possible outcome of a variable is called an event.
- **Sample space:** The collection of all possible events is called sample space.
- **Random variables:** Random variables are used to represent the events and objects in the real world.
- **Prior probability:** The prior probability of an event is probability computed before observing new information.
- **Posterior Probability:** The probability that is calculated after all evidence or information has taken into account. It is a combination of prior probability and new information.

6. Define Conditional probability.

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

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

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

P(B)= Marginal probability of B.

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

Solution:

Let, A is an event that a student likes Mathematics

B is an event that a student likes English.

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

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

8. Define Bayesian Inference.

- Bayesian inference is a probabilistic approach to machine learning that provides estimates of the probability of specific events.
- Bayesian inference is a statistical method for understanding the uncertainty inherent in prediction problems.
- Bayesian inference algorithm can be viewed as a Markov Chain Monte Carlo algorithm that uses prior probability distributions to optimize the likelihood function.

9. List Bayes Theorem or Bayes Rule

- Bayes' theorem can be derived using product rule and conditional probability of event A with known event B:
- **Product Rule:**
  1. P(A ∧ B)= P(A|B) P(B) or
  2. P(A ∧ B)= P(B|A) P(A)
- **Conditional Probability:**
  - Let A and B are events,
  - P(A|B) is the conditional probability of A given B,
  - P(B|A) is the conditional probability of B given A.
- Equating right hand side of both the equations will get:

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

The above equation (a) is called as **Bayes' rule** or **Bayes' theorem**. This equation is basic of most modern AI systems for **probabilistic inference**.

- P(A|B) is known as **posterior**, is the Probability of hypothesis A when occurred an evidence B.
- P(B|A) is called the **likelihood,** in which hypothesis is true, then calculate the probability of evidence.
- P(A) is called the **prior probability**, probability of hypothesis before considering the evidence
- P(B) is called **marginal probability**, pure probability of an evidence.

10. Suppose we want to perceive the effect of some unknown cause, and want to compute that cause, then the Bayes' rule becomes:

$$\mathbf{P(cause\,|\,effect)} = \frac{P(effect|cause)\ P(cause)}{P(effect)}$$

what is the probability that a patient has diseases meningitis with a stiff neck?

Given Data:

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

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

Solution

Let a be the proposition that patient has stiff neck and b be the proposition that patient has meningitis.

So, calculate the following as:

P(a|b) = 0.8

P(b) = 1/30000

P(a)= .02

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

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

11. Consider two events: A (it will rain tomorrow) and B (the sun will shine tomorrow).

- Use Bayes' theorem to compute the posterior probability of each event occurring, given the resulting weather conditions for today:

P(A|sunny) = P(sunny|A) * P(A) / P(sunny)

P(B|sunny) = P(sunny|B) * P(B) / P(sunny)

where sunny is our evidence (the resulting weather condition for today).

12. What are the Application of Bayes' theorem in Artificial intelligence?

- It is used to calculate the next step of the robot when the already executed step is given.
- Bayes' theorem is helpful in weather forecasting.

13. Define 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 Network can be used for building models from data and experts opinions, and it consists of two parts:
  - o Directed Acyclic Graph
  - o Table of conditional probabilities

14. Define Joint probability distribution.

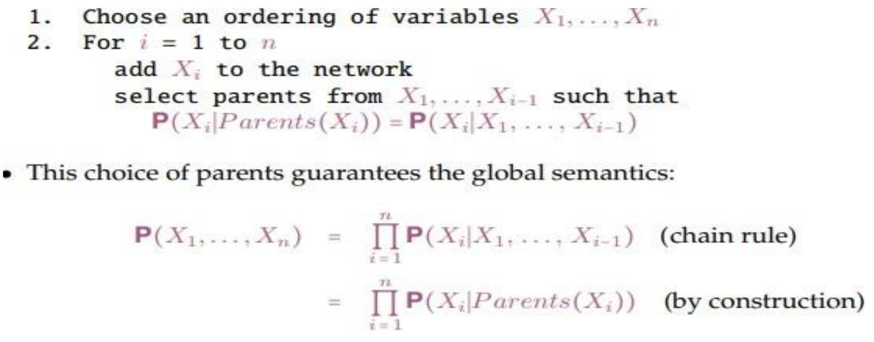
- If variables are x1, x2, x3,....., xn, then the probabilities of a different combination of x1, x2, x3.. xn, are known as Joint probability distribution.
- **P[x_1, x_2, x_3, ,x_n]**, can be written as the following way in terms of the joint probability distribution.

  = P[x_1| x_2, x_3,....., x_n]. p[x_2, x_3, , x_n]

  = P[x_1| x_2, x_3,....., x_n]P[x_2|x_3,....., x_n] P[x_{n-1}|x_n]P[x_n].
- In general for each variable Xi,

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

15. Write an algorithm for Constructing Bayesian Network



```
1.  Choose an ordering of variables $X_1, \ldots, X_n$
2.  For $i$ = 1 to $n$
      add $X_i$ to the network
      select parents from $X_1, \ldots, X_{i-1}$ such that
        $\mathbf{P}(X_i|Parents(X_i)) = \mathbf{P}(X_i|X_1, \ldots, X_{i-1})$
```

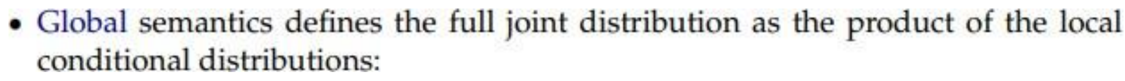
- This choice of parents guarantees the global semantics:

$$\mathbf{P}(X_1,\ldots,X_n) = \prod_{i=1}^{n} \mathbf{P}(X_i|X_1,\ldots,X_{i-1}) \quad \text{(chain rule)}$$

$$= \prod_{i=1}^{n} \mathbf{P}(X_i|Parents(X_i)) \quad \text{(by construction)}$$

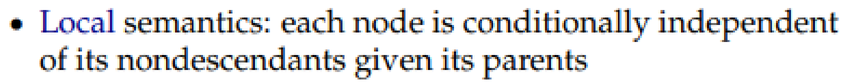
16. Define Global semantics and local semantics.

Global Semantics



- Global semantics defines the full joint distribution as the product of the local conditional distributions:

Local Semantics



- Local semantics: each node is conditionally independent of its nondescendants given its parents

17. List the ways to understand 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.

18. What are the Applications of Bayesian networks in AI?

1. Spam filtering
2. Bio monitoring
3. Information retrieval
4. Image processing
5. Gene regulatory network
6. Turbo code
7. Document classification

19. Define Bayesian Inference.

- Bayesian Network is to perform inference, which computes the marginal probability P(V=v) for each node V and each possible instantiation v.
- Inference can also be done on a Bayesian network when the values of some nodes are known (as evidence) and wish to compute the likelihood of values of other nodes.
- There are two types of inference on Bayesian networks: exact and approximate.
- Exact inference algorithms compute the exact values of each marginal or posterior probability, while approximate inference algorithms sacrifice some accuracy of the probabilities to report results quickly.

20. Define Exact Inference.

- The goal of an exact inference algorithm is to report the exact values for either the marginal (P(V = v)) or posterior probabilities (P(V = v|e)) for each instantiation v of each node V, possible given some evidence e of other node values.

21. List the common exact inference algorithms –

- Pearl's algorithm
- Lauritzen-Spiegelhalter algorithm.

22. Define Pearl's Algorithm.

- Pearl's algorithm is a linear-time algorithm that computes the posterior probabilities of each node given evidence of singly-connected networks.
- Pearl introduced the notation λ (X = x) for the diagnostic support of a node X with value x, which is the probability of evidence below X given that X = x.

23. Define Lauritzen-Spiegelhalter (LS) Algorithm.

- The Lauritzen-Spiegelhalter (LS) algorithm is an inference algorithm for Bayesian networks that works on all models.

24. Define Causal Network or Causal Bayesian Network

- A causal network is an acyclic digraph arising from an evolution of a substitution system, and representing its history.
- In an evolution of a multiway system, each substitution event is a vertex in a causal network.
- Two events which are related by causal dependence, meaning one occurs just before the other, have an edge between the corresponding vertices in the causal network.
- More precisely, the edge is a directed edge leading from the past event to the future event.

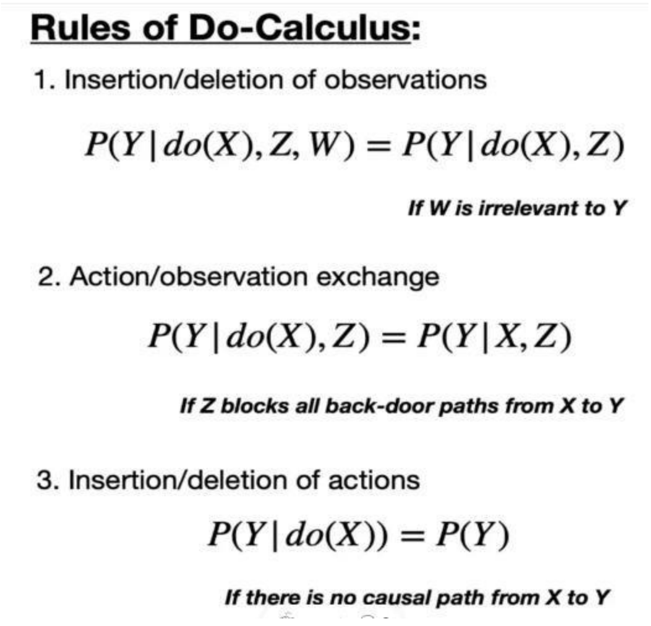
25. Define Structural Causal Models (SCMs).

- SCMs consist of two parts: a graph, which visualizes causal connections, and equations, which express the details of the connections. a **graph** is a **mathematical construction that consists of vertices (nodes) and edges (links)**.
- SCMs use a special kind of graph, called a **Directed Acyclic Graph (DAG)**, for which all edges are directed and no cycles exist.
- DAGs are a common starting place for causal inference.

26. List the purpose of do-operator in causal networks.

- The **do-operator** is a **mathematical representation of a physical intervention**.
- If the model starts with Z → X → Y, simulate an intervention in X by deleting all the incoming arrows to X, and manually setting X to some value x_0.

27. List the rules of Do-Calculus.



Rules of Do-Calculus:

1. Insertion/deletion of observations

$$P(Y|do(X),Z,W) = P(Y|do(X),Z)$$

If W is irrelevant to Y

2. Action/observation exchange

$$P(Y|do(X),Z) = P(Y|X,Z)$$

If Z blocks all back-door paths from X to Y

3. Insertion/deletion of actions

$$P(Y|do(X)) = P(Y)$$

If there is no causal path from X to Y

PART B

1. Explain the concept of uncertainty and acting under uncertainty with suitable example. Explain in detail about probabilistic reasoning.

UNCERTAINITY & PROBABILISTIC REASONING

1.1 Uncertainty:

1.1.1 Causes of uncertainty

1.2 Probabilistic reasoning:

1.2.1 Need of probabilistic reasoning in AI

1.2.2 Ways to solve problems with uncertain knowledge

1.2.3 Probability

1.2.4 Conditional probability

1.2.4.1 Example

Agents almost never have access to the whole truth about their environment. Agents must, therefore, act under **uncertainty.**

Handling uncertain knowledge

In this section, we look more closely at the nature of uncertain knowledge. We will use a simple diagnosis example to illustrate the concepts involved. Diagnosis whether for medicine, automobile repair, or whatever-is a task that almost always involves uncertainty. Let us try to write rules for dental diagnosis using first-order logic, so that we can see how the logical approach breaks down. Consider the following rule:

$$\forall p \;\; Symptom(p, Toothache) \Rightarrow Disease(p, Cavity)\,.$$

The problem is that this rule is wrong. Not all patients with toothaches have cavities; some of them have gum disease, an abscess, or one of several other problems:

$$\forall p \;\; Symptom(p, Toothache) \Rightarrow$$
$$Disease(p, Cavity) \lor Disease(p, GumDisease) \lor Disease(p, Abscess) \ldots$$

Unfortunately, in order to make the rule true, we have to add an almost unlimited list of possible causes. We could try turning the rule into a causal rule:

$$\forall p \;\; Disease(p, Cavity) \Rightarrow Symptom(p, Toothache)\,.$$

But this rule is not right either; not all cavities cause pain The only way to fix the rule is to make it logically exhaustive: to augment the left-hand side with all the qualifications required for a cavity to cause a toothache. Even then, for the purposes of diagnosis, one must also take into account the possibility that the patient might have a toothache and a cavity that are unconnected. Trying to use first-order logic to cope with a domain like medical diagnosis thus fails for three main reasons:

0 **Laziness:** It is too much work to list the complete set of antecedents or consequents needed to ensure an exceptionless rule and too hard to use such rules.
0 **Theoretical ignorance:** Medical science has no complete theory for the domain.
0 **Practical ignorance:** Even if we know all the rules, we might be uncertain about a particular patient because not all the necessary tests have been or can be run.

The connection between toothaches and cavities is just not a logical consequence in either direction. This is typical of the medical domain, as well as most other judgmental domains: law, business, design, automobile repair, gardening, dating, and so on. The agent's knowledge can at best provide only a **degree of belief** in the relevant sentences. Our main tool for dealing with degrees of belief will be **probability theory,** which assigns to each sentence a numerical degree of belief between 0 and 1.

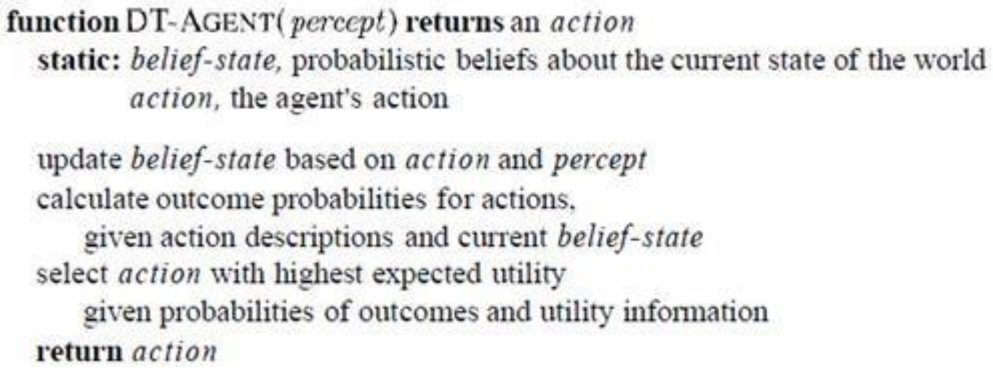
Probability provides a way of **summarizing** *the uncertainty that comes from our laziness and ignorance.* We might not know for sure what afflicts a particular patient, but we believe that there is, say, an 80% chance-that is, a probability of 0.8-that the patient has a cavity if he or she has a toothache.

That is, we expect that out of all the situations that are indistinguishable from the current situation as far as the agent's knowledge goes, the patient will have a cavity in 80% of them. This belief could be derived from statistical data-80% of the toothache patients seen so far have had cavities-or from some general rules, or from a combination of evidence sources.

The 80% summarizes those cases in which all the factors needed for a cavity to cause a toothache are present and other cases in which the patient has both toothache and cavity but the two are unconnected. The missing 20% summarizes all the other possible causes of toothache that we are too lazy or ignorant to confirm or deny.

Design for a decision-theoretic agent

Below algorithm sketches the structure of an agent that uses decision theory to select actions. The agent is identical, at an abstract level, to the logical agent. The primary difference is that the decision-theoretic agent's knowledge of the current state is uncertain; the agent's **belief state** is a representation of the probabilities of all possible actual states of the world. As time passes, the agent accumulates more evidence and its belief state changes. Given the belief state, the agent can make probabilistic predictions of action outcomes and hence select the action with highest expected utility.



```
function DT-AGENT(percept) returns an action
  static: belief-state, probabilistic beliefs about the current state of the world
          action, the agent's action

  update belief-state based on action and percept
  calculate outcome probabilities for actions,
      given action descriptions and current belief-state
  select action with highest expected utility
      given probabilities of outcomes and utility information
  return action
```

1.1 Uncertainty:

- The knowledge representation, A→B, means if A is true then B is true, but a situation where not sure about whether A is true or not then cannot express this statement, this situation is called uncertainty.
- So to represent uncertain knowledge, uncertain reasoning or probabilistic reasoning is used.

1.1.1 Causes of uncertainty:

Causes of uncertainty in the real world

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

1.2 Probabilistic reasoning:

- Probabilistic reasoning is a way of knowledge representation, the concept of probability is applied to indicate the uncertainty in knowledge.

1.2.1 Need of probabilistic reasoning in AI:

- When there are unpredictable outcomes.

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

1.2.2 Ways to solve problems with uncertain knowledge:

- Bayes' rule
- Bayesian Statistics

1.2.3 Probability:

- Probability can be defined as a chance that an uncertain event will occur.
- The value of probability always remains between 0 and 1 that represent ideal uncertainties.
  - $0 \leq P(A) \leq 1$, where P(A) is the probability of an event A.
  - $P(A) = 0$, indicates total uncertainty in an event A.
  - $P(A) = 1$, indicates total certainty in an event A.
- Formula to find the probability of an uncertain event

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

$P(\neg A)$ = probability of a not happening event.

$P(\neg A) + P(A) = 1$.

- **Event:** Each possible outcome of a variable is called an event.
- **Sample space:** The collection of all possible events is called sample space.
- **Random variables:** Random variables are used to represent the events and objects in the real world.
- **Prior probability:** The prior probability of an event is probability computed before observing new information.
- **Posterior Probability:** The probability that is calculated after all evidence or information has taken into account. It is a combination of prior probability and new information.

1.2.4 Conditional probability:

- Conditional probability is a probability of occurring an event when another event has already happened.

- Let's suppose, to calculate the event A when event B has already occurred, "the probability of A under the conditions of B", it is:

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

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

P(B)= Marginal probability of B.

- If the probability of A is given and to find the probability of B, then it is:

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

1.2.4.1 Example:

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

Solution:

Let, A is an event that a student likes Mathematics

B is an event that a student likes English.

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

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

2. Explain in detail about Bayesian inference and Naive Bayes Model or Naive Bayes Theorem or Bayes Rule.

Naive Bayes Model or Naive Bayes Theorem or Bayes Rule

2.1 Bayesian Inference

- Bayesian inference is a probabilistic approach to machine learning that provides estimates of the probability of specific events.
- Bayesian inference is a statistical method for understanding the uncertainty inherent in prediction problems.

- Bayesian inference algorithm can be viewed as a Markov Chain Monte Carlo algorithm that uses prior probability distributions to optimize the likelihood function.
- The basis of Bayesian inference is the notion of apriori and a posteriori probabilities.
    - The priori probability is the probability of an event before any evidence is considered.
    - The posteriori probability is the probability of an event after taking into account all available evidence.
- For example, if we want to know the probability that it will rain tomorrow, our priori probability would be based on our knowledge of the weather patterns in our area.

2.2 Bayes Theorem or Bayes Rule

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

3. P(A ∧ B)= P(A|B) P(B) or
4. P(A ∧ B)= P(B|A) P(A)

- **Conditional Probability:**
    - Let A and B are events,
    - P(A|B) is the conditional probability of A given B,
    - P(B|A) is the conditional probability of B given A.
- Equating right hand side of both the equations will get:

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

The above equation (a) is called as **Bayes' rule** or **Bayes' theorem**. This equation is basic of most modern AI systems for **probabilistic inference**.

- P(A|B) is known as **posterior**, is the Probability of hypothesis A when occurred an evidence B.
- P(B|A) is called the **likelihood,** in which hypothesis is true, then calculate the probability of evidence.
- P(A) is called the **prior probability**, probability of hypothesis before considering the evidence
- P(B) is called **marginal probability**, pure probability of an evidence.
- In general,

P (B) = P(A)*P(B|Ai),

- Hence the Bayes' rule can be written as:

$$P(A_i|B) = \frac{P(A_i)*P(B|A_i)}{\sum_{i=1}^{k} P(A_i)*P(B|A_i)}$$

Where A_1, A_2, A_3,........, A_n is a set of mutually exclusive and exhaustive events.

2.3 Example 1 - Applying Bayes' rule:

Suppose we want to perceive the effect of some unknown cause, and want to compute that cause, then the Bayes' rule becomes:

$$\textbf{P(cause|effect)} = \frac{P(effect|cause)\ P(cause)}{P(effect)}$$

what is the probability that a patient has diseases meningitis with a stiff neck?

Given Data:

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

The Known probability that a patient has meningitis disease is 1/30,000.

The Known probability that a patient has a stiff neck is 2%.

Solution

Let a be the proposition that patient has stiff neck and b be the proposition that patient has meningitis.

So, calculate the following as:

$P(a|b) = 0.8$

$P(b) = 1/30000$

$P(a)= .02$

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

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

Example 2 - Applying Bayes' rule:

- Consider two events: A (it will rain tomorrow) and B (the sun will shine tomorrow).
- Use Bayes' theorem to compute the posterior probability of each event occurring, given the resulting weather conditions for today:

  P(A|sunny) = P(sunny|A) * P(A) / P(sunny)

  P(B|sunny) = P(sunny|B) * P(B) / P(sunny)

  where sunny is our evidence (the resulting weather condition for today).
- From these equations,
  - if event A is more likely to result in sunny weather than event B, then the posterior probability of A occurring, given that the resulting weather condition for today is sunny, will be higher than the posterior probability of B occurring.
  - Conversely, if event B is more likely to result in sunny weather than event A, then the posterior probability of B occurring, given that the resulting weather condition for today is sunny, will be higher than the posterior probability of A occurring.

2.4 Application of Bayes' theorem in Artificial intelligence:

- It is used to calculate the next step of the robot when the already executed step is given.
- Bayes' theorem is helpful in weather forecasting.

Naive Bayes Theorem

The dentistry example illustrates a commonly occurring pattern in which a single cause directly influences a number of effects, all of which are conditionally independent, given the cause. The full joint distribution can be written as

$$\mathbf{P}(Cause, Effect_1, \ldots, Effect_n) = \mathbf{P}(Cause) \prod_i \mathbf{P}(Effect_i \mid Cause) .$$

Such a probability distribution is called a naive Bayes model—"naive" because it is often used (as a simplifying assumption) in cases where the "effect" variables are not strictly independent given the cause variable. (The naive Bayes model is sometimes called a Bayesian classifier, a somewhat careless usage that has prompted true Bayesians to call it the idiot Bayes model.) In practice, naive Bayes systems often work very well, even when the conditional independence assumption is not strictly true

3. Explain in detail about Bayesian Network

Bayesian Network

3.1 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 Network can be used for building models from data and experts opinions,and it consists of two parts:
  - Directed Acyclic Graph
  - Table of conditional probabilities
- The generalized form of Bayesian network that represents and solve decisionproblems under uncertain knowledge is known as an Influence diagram.
- It is used to represent conditional dependencies.
- It can also be used in various tasks including prediction, anomaly detection, diagnostics, automated insight, reasoning, time series prediction, and decision making under uncertainty.
- A Bayesian network graph is made up of nodes and Arcs (directed links).

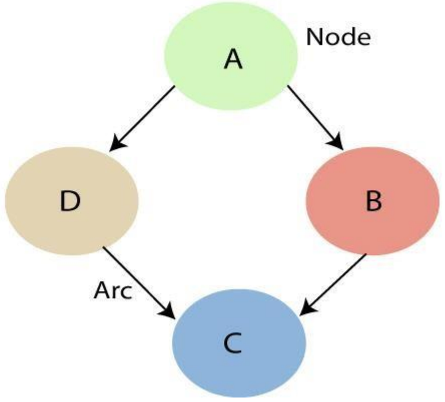


Figure 2.1 – Example for Bayesian Network

- Each node corresponds to the random variables, and a variable can be continuous or discrete.

- Arc or directed arrows represent the causal relationship or conditional probabilities between random variables.
- These directed links or arrows connect the pair of nodes in the graph.
- These links represent that one node directly influence the other node, and if there is no directed link that means that nodes are independent with each other.

  Example

  In the figure 2.1, A, B, C, and D are random variables represented by the nodes of the network graph.

  - 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 graph does not contain any cyclic graph. Hence, it is known as a directed acyclic graph or DAG.
- The Bayesian network has mainly two components:
  1. Causal Component
  2. Actual numbers

- Each node in the Bayesian network has condition probability distribution $\mathbf{P(X_i\ |Parent(X_i)\ \ )}$, which determines the effect of the parent on that node.
- Bayesian network is based on Joint probability distribution and conditional probability.

3.2 Joint probability distribution:

- If variables are x1, x2, x3,....., xn, then the probabilities of a different combination of x1, x2, x3.. xn, are known as Joint probability distribution.
- $\mathbf{P[x_1, x_2, x_3, \quad ,x_n]}$, can be written as the following way in terms of the joint probability distribution.

  $$\mathbf{= P[x_1|\ x_2, x_3,\ldots.., x_n].\ p[x_2, x_3, \quad , x_n]}$$

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

- In general for each variable Xi,

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

3.3 Constructing Bayesian Network

```
1. Choose an ordering of variables $X_1, \ldots, X_n$
2. For $i$ = 1 to $n$
     add $X_i$ to the network
     select parents from $X_1, \ldots, X_{i-1}$ such that
        $\mathbf{P}(X_i|Parents(X_i)) = \mathbf{P}(X_i|X_1, \ldots, X_{i-1})$
```

- This choice of parents guarantees the global semantics:

$$\begin{aligned}\mathbf{P}(X_1, \ldots, X_n) &= \prod_{i=1}^{n} \mathbf{P}(X_i|X_1, \ldots, X_{i-1}) \quad \text{(chain rule)}\\ &= \prod_{i=1}^{n} \mathbf{P}(X_i|Parents(X_i)) \quad \text{(by construction)}\end{aligned}$$

Global Semantics

- Global semantics defines the full joint distribution as the product of the local conditional distributions:

$$P(x_1, \ldots, x_n) = \prod_{i=1}^{n} P(x_i|parents(X_i))$$

Local Semantics

- Local semantics: each node is conditionally independent of its nondescendants given its parents

Markov Blanket

- Each node is conditionally independent of all others given its **Markov blanket:** parents + children + children's parents

3.4 Example:

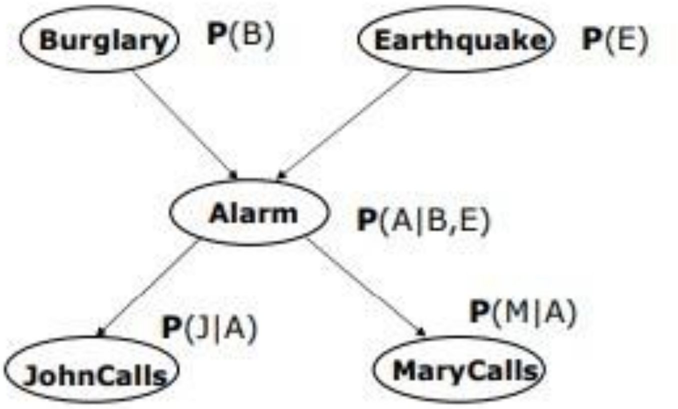
Harry installed a new burglar alarm at his home to detect burglary. The alarm reliably responds at detecting a burglary but also responds for minor earthquakes. Harry has two neighbors David and Sophia, who have taken a responsibility to inform Harry at work when they hear the alarm. David always calls 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.

Solution:

- The Bayesian network for the above problem is given in figure 2.2. 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.

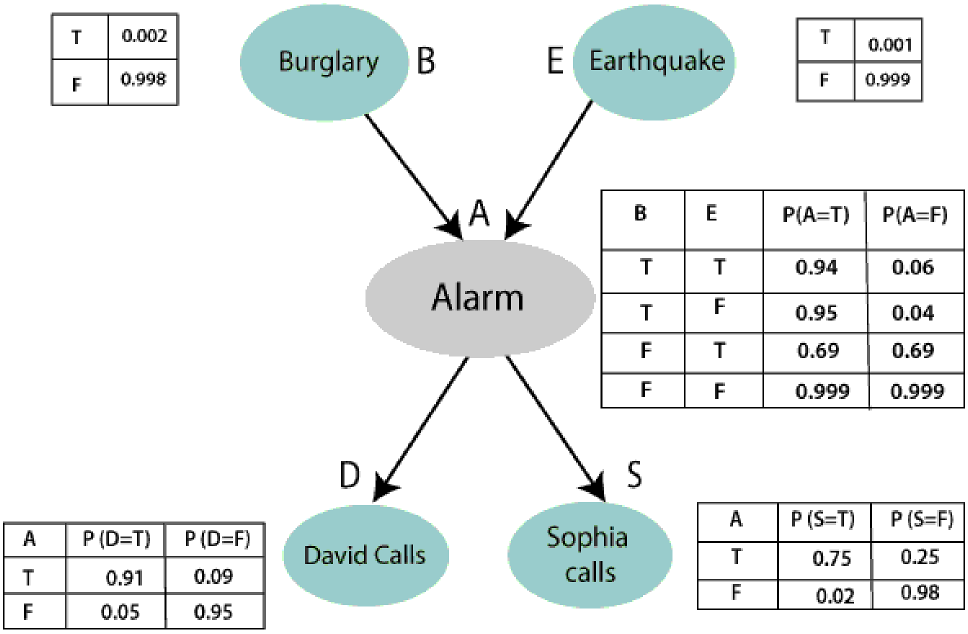


Variables:

Burglar, Earthquake, Alarm, John Calls, Mary Calls

Network topology reflects "causal" knowledge

- A burglar can set the alarm off
- An earthquake can set the alarm off
- The alarm can cause Mary to call
- The alarm can cause John to call



T	0.002
F	0.998

T	0.001
F	0.999

B	E	P(A=T)	P(A=F)
T	T	0.94	0.06
T	F	0.95	0.04
F	T	0.69	0.69
F	F	0.999	0.999

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

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

Figure 2.2 - The Bayesian network for the example problem

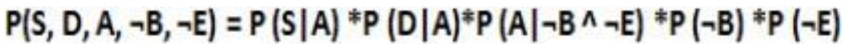
All events occurring in this network:

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

Write the events of problem statement in the form of probability:

P[D, S, A, B, E],

Rewrite the probability statement using joint probability distribution:



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

Let's take the observed probability for the Burglary and earthquake component:

- P(B=True) = 0.002, which is the probability of burglary.
- P(B=False)= 0.998, which is the probability of no burglary.
- P(E=True)= 0.001, which is the probability of a minor earthquake
- P(E=False)= 0.999, Which is the probability that an earthquake not occurred.

Conditional probability table for Alarm A:

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

B	E	P(A= True)	P(A= False)
True	True	0.94	0.06
True	False	0.95	0.04
False	True	0.31	0.69
False	False	0.001	0.999

Conditional probability table for David Calls:

The Conditional probability of David that he will call depends on the probability of Alarm.

A	P(D= True)	P(D= False)
True	0.91	0.09
False	0.05	0.95

Conditional probability table for Sophia Calls:

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

A	P(S= True)	P(S= False)
True	0.75	0.25
False	0.02	0.98

From the formula of joint distribution, 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.

3.5 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.

3.6 Applications of Bayesian networks in AI

Bayesian networks find applications in a variety of tasks such as:

1. **Spam filtering:**
   a. A spam filter is a program that helps in detecting unsolicited and spam mails. Bayesian spam filters check whether a mail is spam or not.

2. **Biomonitoring:**
   a. This involves the use of indicators to quantify the concentration of chemicals in the human body.
3. **Information retrieval:**
   a. Bayesian networks assist in information retrieval for research, which is a constant process of extracting information from databases.
4. **Image processing:**
   a. A form of signal processing, image processing uses mathematical operations to convert images into digital format.
5. **Gene regulatory network**:
   a. A Bayesian network is an algorithm that can be applied to gene regulatory networks in order to make predictions about the effects of genetic variations on cellular phenotypes.
   b. Gene regulatory networks are a set of mathematical equations that describe the interactions between genes, proteins, and metabolites.
   c. They are used to study how genetic variations affect the development of a cell or organism.
6. **Turbo code:**
   a. Turbo codes are a type of error correction code capable of achieving very high data rates and long distances between error correcting nodes in a communications system.
   b. They have been used in satellites, space probes, deep-space missions, military communications systems, and civilian wireless communication systems, including WiFi and 4G LTE cellular telephone systems.
7. **Document classification:**
   a. The main issue is to assign a document multiple classes. The task can be achieved manually and algorithmically. Since manual effort takes too much time, algorithmic documentation is done to complete it quickly and effectively.

4. Explain in detail about Bayesian Inference and its type Exact Inference with suitable example.

Exact inference in Bayesian networks

The basic task for any probabilistic inference system is to compute the posterior probability distribution for a set of query variables, given some observed event-that is, some assignment of values to a set of evidence variables. We will use the notation X denotes the query variable; E denotes the set of evidence variables El, . . . , *Em,* and e is a particular observed event; Y will denote the nonevidence variables Yl, . . . , (some- times called the hidden variables). Thus, the complete set of variables X = {X} U E U Y. A typical query asks for the posterior probability distribution P(X|e)4

In the burglary network, we might observe the event in which *JohnCalls* = *true* and *MaryCalls* = *true.* We could then ask for, say, the probability that a burglary has occurred:

$$\mathbf{P}(Burglary|JohnCalls=true, MaryCalls=true) = \langle 0.284, 0.716\rangle \,.$$

Inference by enumeration

Conditional probability can be computed by summing terms from the full joint distribution. More specifically, a query P(X|e) can be answered using Equation, which we repeat here for convenience:

$$\mathbf{P}(X|\mathbf{e}) = \alpha\,\mathbf{P}(X,\mathrm{e}) = \alpha \sum_{\mathbf{Y}} \mathbf{P}(X,\mathrm{e},\mathrm{y})\,.$$

Now, a Bayesian network gives a complete representation of the full joint distribution. More specifically, Equation shows that the terms *P(x,* e, y) in the joint distribution can be written as products of conditional probabilities from the network. Therefore, *a query can be answered using a Bayesian network by computing sums of products of conditional probabibities from the network.* In Figure an algorithm, ENUMERATE-JOINT-ASK, was given for inference by enumeration from the full joint distribution. The algorithm takes as input a full joint distribution P and looks up values therein. It is a simple matter to modify the algorithm so that it takes as input a Bayesian network bn and "looks up" joint entries by multiplying the corresponding CPT entries from bn.

Consider the query P(Burglary1 JohnCalls = true, &Jury Calls = true). The hidden variables for this query are Earthquake and Alarm. From Equation (13.6), using initial letters for the variables in order to shorten the expressions, we have

$$\mathbf{P}(B|j,m) = \alpha\,\mathbf{P}(B,j,m) = \alpha \sum_{e}\sum_{a} \mathbf{P}(B,e,a,j,m)\,.$$

The semantics of Bayesian networks then gives us an expression in terms of CPT entries. For simplicity, we will do this just for Burglizry = true:

$$P(b|j,m) = \alpha \sum_{e}\sum_{a} P(b)P(e)P(a|b,e)P(j|a)P(m|a)$$

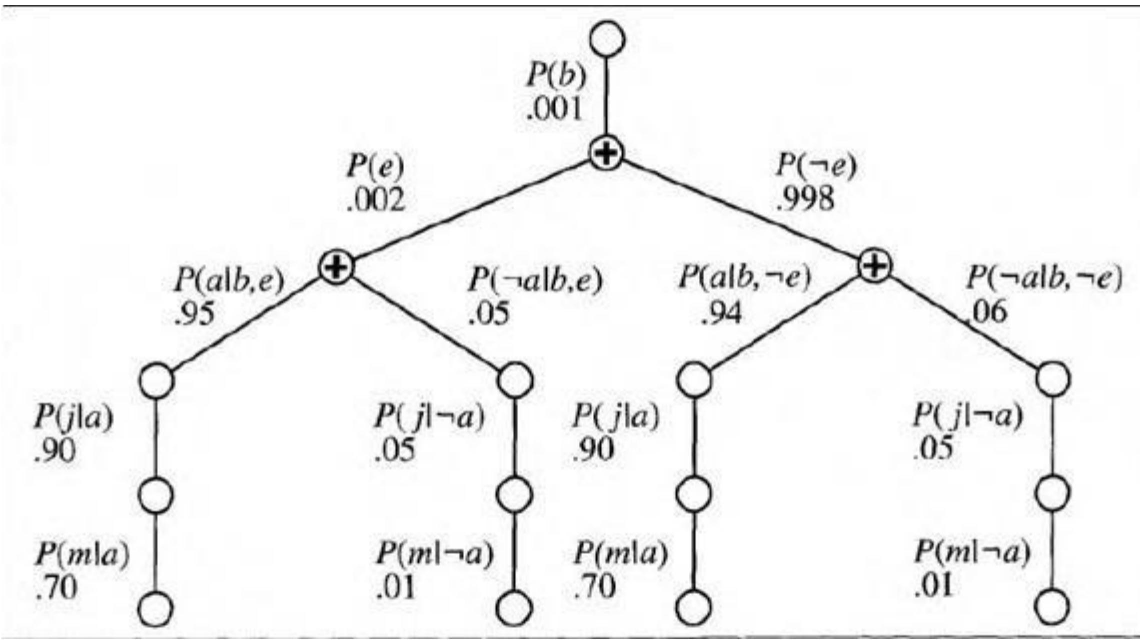
To compute this expression, we have to add four terms, each computed by multiplying five numbers. In the worst case, where we have to sum out almost all the variables, the complexity of the algorithm for a network with n Boolean variables is *O(n2n).* An improvement can be obtained from the following simple observations: the P(b) term is a constant and can be moved outside the summaltions over a and e, and the *13(e)* term can be moved outside the summation over a. Hence, we have

$$P(b|j,m) = \alpha P(b) \sum_{e} P(e) \sum_{a} P(a|b,e) P(j|a) P(m|a) \,.$$

This expression can be evaluated by looping through the variables in order, multiplying CPT entries as we go. For each summation, we also need to loop over the variable's possible values. The structure of this computation is shown in Figure. Using the numbers from Figure, we obtain P(b| j , m) = *a* x 0.00059224. 'The correspondingc omputation for ~b yields *a* x 0.0014919; hence

$$\mathbf{P}(B|j,m) = \alpha \langle 0.00059224, 0.0014919 \rangle \approx \langle 0.284, 0.716 \rangle \,.$$

This expression can be evaluated by looping through the variables in order, multiplying CPT entries as we go. For each summation, we also need to loop over the variable's possible values.



The structure of this computation is shown in above Figure. Using the numbers from Figure, we obtain P(b| j , m) = *a* x 0.00059224. 'The co~respondingc omputation for ~b yields *a* x 0.0014919; hence

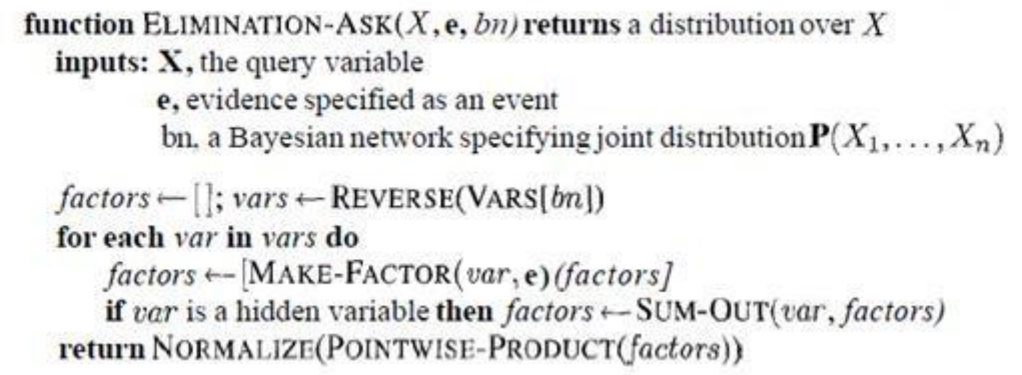
$$\mathbf{P}(B|j,m) = \alpha \langle 0.00059224, 0.0014919 \rangle \approx \langle 0.284, 0.716 \rangle \,.$$

That is, the chance of a burglary, given calls from both neighbors, is about 28%. The evaluation process for the expression in Equation is shown as an expression tree in Figure.

The variable elimination algorithm

The enumeration algorithm can be improved substantially by eliminating repeated calculations of the kind illustrated in Figure. The idea is simple: do the calculation once and save the results for later use. This is a form of dynamic programming. There are several versions of this approach; we present the variable elimination algorithm, which is the simplest. Variable elimination works by evaluating expressions such as Equation in *right-to-left* order (that is, *bottom-up* in Figure). Intermediate results are stored, and summations over each variable are done only for those portions of the expression that depend on the variable. Let us illustrate this process for the burglary network. We evaluate the expression

$$\mathbf{P}(B|j,m) = \alpha \underbrace{\mathbf{P}(B)}_{B} \sum_{e} \underbrace{P(e)}_{E} \sum_{a} \underbrace{\mathbf{P}(a|B,e)}_{A} \underbrace{P(j|a)}_{J} \underbrace{P(m|a)}_{M} .$$



```
function ELIMINATION-ASK(X, e, bn) returns a distribution over X
  inputs: X, the query variable
          e, evidence specified as an event
          bn. a Bayesian network specifying joint distribution $\mathbf{P}(X_1, \ldots, X_n)$

  factors ← [ ]; vars ← REVERSE(VARS[bn])
  for each var in vars do
      factors ← [MAKE-FACTOR(var, e)(factors]
      if var is a hidden variable then factors ← SUM-OUT(var, factors)
  return NORMALIZE(POINTWISE-PRODUCT(factors))
```

The complexity of exact inference

We have argued that variable elimination is more efficient than enumeration because it avoids repeated computations (as well as dropping irrelevant variables). The time and space requirements of variable elimination are dominated by the size of the largest factor constructed during the operation of the algorithm. This in turn is determined by the order of elimination of variables and by the structure of the network.

The burglary network of Figure belongs to the family of networks in which there is at most one undirected path between any two nodes in the network. These are called **singly connected** networks or **polytrees,** and they have a particularly nice property: *The time and space complexity of exact inference in polytrees is linear in the size of the network.* I-Iere, the size is defined as the number of CPT entries; if the number of parents of each node is bounded by a constant, then the complexity will also be linear in the number of nodes. These results hold for any ordering consistent with the topological ordering of the network .

For **multiply connected** networks, such as that of Figure, variable elimination can have exponential time and space complexity in the worst case, even when the number of parents per node is bounded. This is not surprising when one considers that, *because it includes inference in propositional logic as a special case, inference in Bayesian networks is 1W-hard.* In fact, it can be shown that the problem is as hard as that of computing the *number* of satisfying assignments for a propositional logic formula. This means that it is #P-hard ("number-P hard")-that is, strictly harder than NP-complete problems.

There is a close connection between the complexity of Bayesian network inference and the complexity of constraint satisfaction problems (CSPs), the difficulty of solving a discrete CSP is related to how "tree-lilce" its constraint graph is Measures such as **hypertree width,** which bound the complexity of solving a CSP, can also be applied directly to Bayesian networks. Moreover, the variable elimination algorithm can be generalized to solve CSPs as well as Bayesian networks.

5. Explain Causal Network or Causal Bayesian Network in Machine

5.1 Causal Network or Causal Bayesian Network

- A causal network is an acyclic digraph arising from an evolution of a substitution system, and representing its history.
- In an evolution of a multiway system, each substitution event is a vertex in a causal network.
- Two events which are related by causal dependence, meaning one occurs just before the other, have an edge between the corresponding vertices in the causal network.
- More precisely, the edge is a directed edge leading from the past event to the future event.
- Refer Figure 2.3 for an example causal network.
- A CBN is a graph formed by nodes representing random variables, connected by links denoting causal influence.

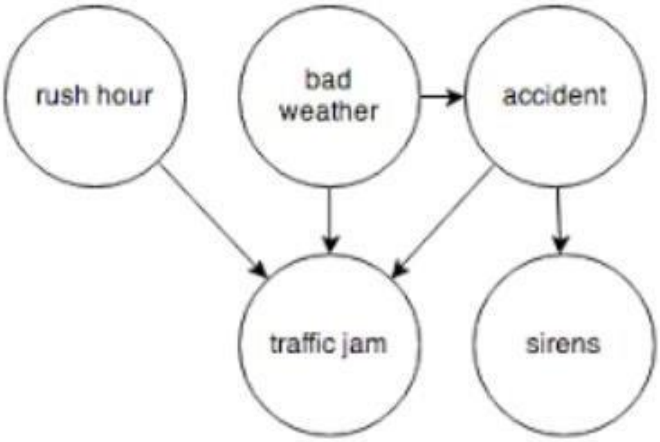


Figure 2.3 – Causal Network Example

- Some causal networks are independent of the choice of evolution, and these are called causally invariant.

- **Structural Causal Models (SCMs)**.
  - SCMs consist of two parts: a graph, which visualizes causal connections, and equations, which express the details of the connections. a **graph** is a **mathematical construction that consists of vertices (nodes) and edges (links)**.
  - SCMs use a special kind of graph, called a **Directed Acyclic Graph (DAG)**, for which all edges are directed and no cycles exist.
  - DAGs are a common starting place for causal inference.
  - Bayesian and causal networks are completely identical. However, the difference lies in their interpretations.

Fire -> Smoke

Bayesian: $P(Smoke \mid Fire)$

Causal: Fire *causes* smoke

- A network with 2 nodes (fire icon and smoke icon) and 1 edge (arrow pointing from fire to smoke).
- This network can be both a Bayesian or causal network.
- The key distinction, however, is when interpreting this network.
- For a **Bayesian** network, we view the **nodes as variables** and the **arrow as a conditional probability**, namely the probability of smoke given information about fire.
- When interpreting this as a **causal** network, we still view **nodes as variables**, however, the **arrow indicates a causal connection**.

- In this case, both interpretations are valid. However, if we were to flip the edge direction, the causal network interpretation would be invalid, since smoke does not *cause* fire.

Implementing Causal Inference

1.The do-operator

- The **do-operator** is a **mathematical representation of a physical intervention**.
- If the model starts with Z → X → Y, simulate an intervention in X by deleting all the incoming arrows to X, and manually setting X to some value x_0. Refer Figure 2.4 denotes the example of do-operator.

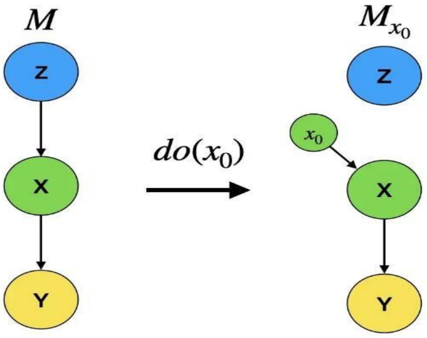
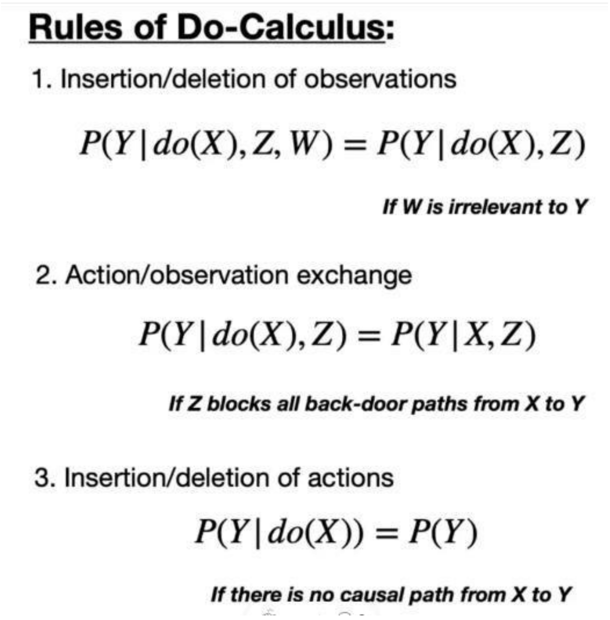


Figure 2.4 – do-operator Example



Rules of Do-Calculus:

1. Insertion/deletion of observations

$$P(Y \mid do(X), Z, W) = P(Y \mid do(X), Z)$$

If W is irrelevant to Y

2. Action/observation exchange

$$P(Y \mid do(X), Z) = P(Y \mid X, Z)$$

If Z blocks all back-door paths from X to Y

3. Insertion/deletion of actions

$$P(Y \mid do(X)) = P(Y)$$

If there is no causal path from X to Y

P(Y|X) is the conditional probability that is, the **probability of Y given an observation of X**. While, **P(Y|do(X))** is the **probability of Y given an *intervention* in X**.

2: Confounding

A simple example of confounding is shown in the figure 2.5 below.

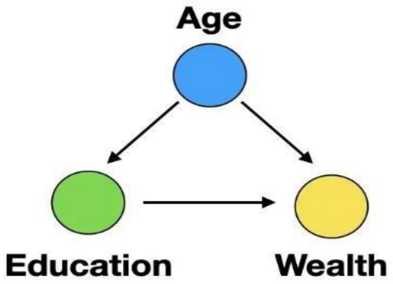


Figure 2.5 – Confounding Example

- In this example, age is a confounder of education and wealth. In other words, if trying to evaluate the impact of education on wealth one would need to *adjust* for age.
- **Adjusting for** (or conditioning on) **age** just means that when looking at age, education, and wealth data, one would compare data points **within** age groups, **not between** age groups.
- Confounding **is** anything that leads to P(Y|X) being different than P(Y|do(X))**.**

3: Estimating Causal Effects

- Treatment effect = (Outcome under E) minus (Outcome under C), that is the difference between the outcome a child would receive if assigned to treatment E and the outcome that same child would receive if assigned to treatment C. These are called potential outcomes.

6. Explain approximate inference in Bayesian network (BN)

Given the intractability of exact inference in large networks, we will now consider approximate inference methods. This section describes randomized sampling algorithms, also called Monte Carlo algorithms, that provide approximate answers whose accuracy depends on the number of samples generated.

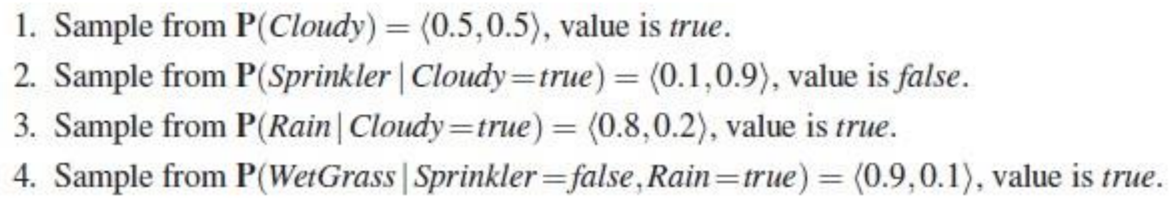
They work by generating random events based on the probabilities in the Bayes net and counting up the different answers found in those random events. With enough samples, we can get arbitrarily close to recovering the true probability distribution—provided the Bayes net has no deterministic conditional distributions

Direct sampling methods

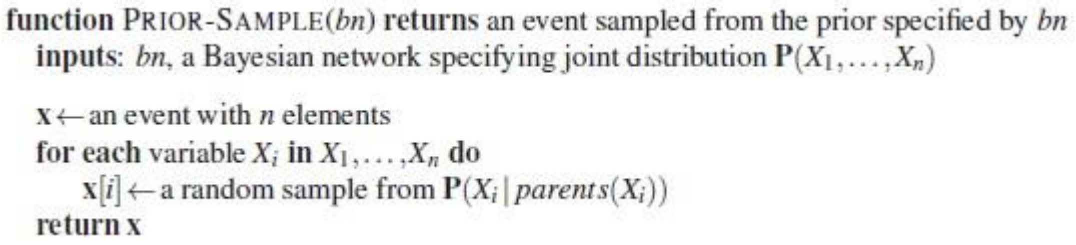
The primitive element in any sampling algorithm is the generation of samples from a known probability distribution. For example, an unbiased coin can be thought of as a random variable *Coin* with values (*heads*, *tails*) and a prior distribution P(*Coin*) = (0.5,0.5). Sampling from this distribution is exactly like flipping the coin: with probability 0.5 it will return *heads*, and with probability 0.5 it will return *tails*.

Given a source of random numbers *r* uniformly distributed in the range [0,1], it is a simple matter to sample any distribution on a single variable, whether discrete or continuous. This is done by constructing the cumulative distribution for the variable and returning the first value whose cumulative probability exceeds *r*

We begin with a random sampling process for a Bayes net that has no evidence associated with it. The idea is to sample each variable in turn, in topological order. The probability distribution from which the value is sampled is conditioned on the values already assigned to the variable's parents. (Because we sample in topological order, the parents are guaranteed to have values already.) This algorithm is shown in Figure. Applying it to the network with the ordering *Cloudy*, *Sprinkler*, *Rain*, *WetGrass*, we might produce a random event as follows:



1. Sample from $\mathbf{P}(Cloudy) = \langle 0.5, 0.5 \rangle$, value is *true*.
2. Sample from $\mathbf{P}(Sprinkler \mid Cloudy = true) = \langle 0.1, 0.9 \rangle$, value is *false*.
3. Sample from $\mathbf{P}(Rain \mid Cloudy = true) = \langle 0.8, 0.2 \rangle$, value is *true*.
4. Sample from $\mathbf{P}(WetGrass \mid Sprinkler = false, Rain = true) = \langle 0.9, 0.1 \rangle$, value is *true*.



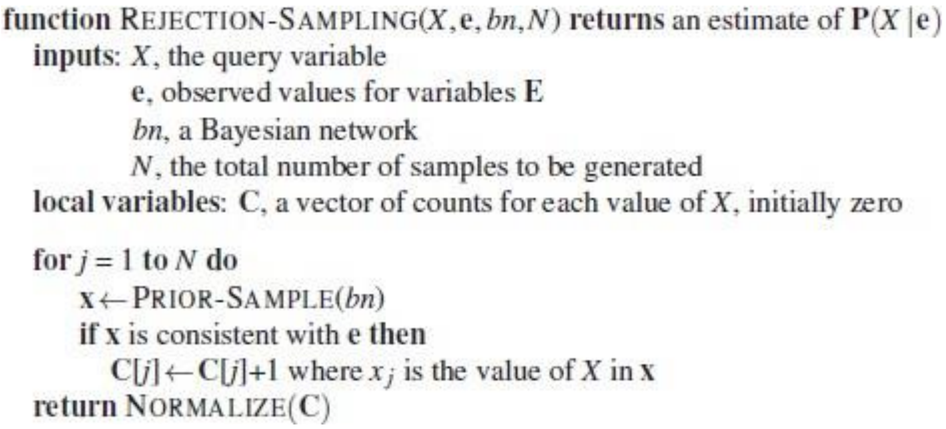
```
function PRIOR-SAMPLE(bn) returns an event sampled from the prior specified by bn
  inputs: bn, a Bayesian network specifying joint distribution $\mathbf{P}(X_1, \ldots, X_n)$

  x ← an event with n elements
  for each variable $X_i$ in $X_1, \ldots, X_n$ do
      x[i] ← a random sample from $\mathbf{P}(X_i \mid parents(X_i))$
  return x
```

Rejection sampling in Bayesian networks

Rejection sampling is a general method for producing samples from a hard-to-sample distribution given an easy-to-sample distribution. In its simplest form, it can be used to compute conditional probabilities that is, to determine *P*(*X* |e). The REJECTION-SAMPLING algorithm is shown in Figure. First, it generates samples from the prior distribution specified by the network. Then, it rejects all those that do not match the evidence. Finally, the estimateˆ*P* (*X* =*x*|e) is obtained by counting how often *X* =*x* occurs in the remaining samples.

Let ˆP(*X* |e) be the estimated distribution that the algorithm returns; this distribution is computed by normalizing N*PS*(*X*,e), the vector of sample counts for each value of *X* where the sample agrees with the evidence e:

$$\hat{\mathbf{P}}(X \mid \mathbf{e}) = \alpha \mathbf{N}_{PS}(X, \mathbf{e}) = \frac{\mathbf{N}_{PS}(X, \mathbf{e})}{N_{PS}(\mathbf{e})} .$$



```
function REJECTION-SAMPLING($X$, e, $bn$, $N$) returns an estimate of $\mathbf{P}(X \mid \mathbf{e})$
   inputs: $X$, the query variable
           e, observed values for variables E
           $bn$, a Bayesian network
           $N$, the total number of samples to be generated
   local variables: C, a vector of counts for each value of $X$, initially zero

   for $j$ = 1 to $N$ do
      x ← PRIOR-SAMPLE($bn$)
      if x is consistent with e then
         C[$j$] ← C[$j$]+1 where $x_j$ is the value of $X$ in x
   return NORMALIZE(C)
```

Inference by Markov chain simulation

In this section, we describe the **Markov chain Monte Carlo** (MCMC) algorithm for inference in Bayesian networks. We will first describe what the algorithm does, then we will explain why it works and why it has such a complicated name.

The MCMC algorithm

MCMC generates each event by making a random change to the preceding event. It is therefore helpful to think of the network as being in a particular *current state* specifying a value for every variable. The next state is generated by randomly sampling a value for one of the nonevidence variables ***Xi,****conditioned on the current values of the variables in the Markov blanket of Xi.* MCMC therefore wanders randomly around the state space-the space of possible complete assignments-flipping one variable at a time, but keeping the evidence variables fixed.

Consider the query *P(Rain1 Sprinkler = true, Wet Grass = true)* applied to the network in Figure. The evidence variables *Sprinkler* and *WetGrass* are fixed to their observed values and the hidden variables *Cloudy* and *Rain* are initialized randomly-let us say to *true* and *false* respectively. Thus, the initial state is *[true, true, false, true].* Now the following steps are executed repeatedly:

1. *Cloudy* is sampled, given the current values of its Markov blanket variables: in this case, we sample from *P(Cloudy1 Sprinkler = true, Rain =false).* Suppose the result is *Cloudy =false.* Then the new current state is *[false, true, false, true].*

2. *Rain* is sampled, given the current values of its Markov blanket variables: in this case, we sample from *P(Rain1 Cloudy =false, Sprinkler = true, WetGrass = true).* Suppose this yields *Rain = true.* The new current state is *[false, true, true, true].*
Each state visited during this process is a sample that contributes to the estimate for the query variable *Rain.* If the process visits 20 states where *Rain* is true and 60 states where *Rain* is false, then the answer to the query is NORMALIZE