**8.Implement a Bayesian network from a given data and infer the data from that Bayesian network.**

A **Bayesian Network** (BN) is a graphical model that represents a set of variables and their probabilistic dependencies using a directed acyclic graph (DAG). In a Bayesian Network:

* **Nodes** represent random variables.
* **Edges** represent conditional dependencies between variables.
* **Conditional probability tables (CPTs)** specify the probability of each node, given its parent nodes.

Bayesian Networks are useful for reasoning under uncertainty, as they enable efficient computation of joint probability distributions.

### Example: Burglary, Earthquake, Alarm, John Calls, David Calls

Let's consider a simple example involving 5 variables:

1. **Burglary (B)**: Whether a burglary occurs.
2. **Earthquake (E)**: Whether an earthquake occurs.
3. **Alarm (A)**: Whether an alarm goes off, which depends on Burglary and Earthquake.
4. **John Calls (J)**: Whether John calls the police if the alarm goes off.
5. **David Calls (D)**: Whether David calls the police if the alarm goes off.

The Bayesian Network can be represented as:

markdown

Copy code

Burglary Earthquake

\ /

\ /

Alarm

/ \

John David

### Relationships:

* **Alarm (A)** depends on both Burglary (B) and Earthquake (E).
* **John Calls (J)** depends on Alarm (A).
* **David Calls (D)** depends on Alarm (A).

### Structure:

* **B** and **E** are independent.
* **A** depends on both **B** and **E**.
* **J** and **D** depend on **A**.

### Conditional Probability Tables (CPTs):

We need to define the probabilities for each node based on its parents.

1. **P(B)**: Probability of Burglary.
   * P(B = True) = 0.001
   * P(B = False) = 0.999
2. **P(E)**: Probability of Earthquake.
   * P(E = True) = 0.002
   * P(E = False) = 0.998
3. **P(A | B, E)**: Probability of Alarm given Burglary and Earthquake.
   * P(A = True | B = True, E = True) = 0.95
   * P(A = True | B = True, E = False) = 0.94
   * P(A = True | B = False, E = True) = 0.29
   * P(A = True | B = False, E = False) = 0.001
4. **P(J | A)**: Probability of John calling given the Alarm.
   * P(J = True | A = True) = 0.90
   * P(J = True | A = False) = 0.05
5. **P(D | A)**: Probability of David calling given the Alarm.
   * P(D = True | A = True) = 0.70
   * P(D = True | A = False) = 0.01

### Inference in Bayesian Networks

The goal of inference is to compute the probability distribution of a certain variable given evidence about other variables. This can be done using algorithms like:

* **Variable Elimination**: Efficiently computes marginal probabilities by eliminating variables through summation.
* **Belief Propagation**: Used for tree-structured Bayesian Networks, propagating beliefs (probabilities) through the network.

### Algorithm: Variable Elimination Example

Let's assume we want to compute the probability that a burglary has occurred given that both John and David called (i.e., P(B | J = True, D = True)).

Steps in **Variable Elimination**:

1. **Set evidence**: We know that both John and David called, so we set J = True and D = True as evidence.
2. **Marginalization**: We eliminate variables that are not of interest (in this case, Earthquake and Alarm) by summing over their possible values.
3. **Update the conditional probabilities**: Using the evidence, update the probabilities and compute the posterior probability P(B | J = True, D = True).

### Computation Example:

We start by expressing the full joint probability distribution:

P(B,E,A,J,D)=P(B)P(E)P(A∣B,E)P(J∣A)P(D∣A)P(B, E, A, J, D) = P(B) P(E) P(A | B, E) P(J | A) P(D | A)P(B,E,A,J,D)=P(B)P(E)P(A∣B,E)P(J∣A)P(D∣A)

Given evidence J=True,D=TrueJ = \text{True}, D = \text{True}J=True,D=True, we sum over the unobserved variables (Earthquake and Alarm) to compute the posterior probability of Burglary.

Let’s manually compute the terms:

P(B=True∣J=True,D=True)∝P(J=True,D=True∣B=True)P(B=True)P(B = \text{True} | J = \text{True}, D = \text{True}) \propto P(J = \text{True}, D = \text{True} | B = \text{True}) P(B = \text{True})P(B=True∣J=True,D=True)∝P(J=True,D=True∣B=True)P(B=True)

!pip install pgmpy

import numpy as np *# linear algebra*

import pandas as pd *# data processing, CSV file I/O (e.g. pd.read\_csv)*

*# Input data files are available in the read-only "../input/" directory*

*# For example, running this (by clicking run or pressing Shift+Enter) will list all files under the input directory*

import os

for dirname, \_, filenames **in** os.walk('/kaggle/input'):

for filename **in** filenames:

print(os.path.join(dirname, filename))

import pgmpy.models

import pgmpy.inference

import networkx as nx

import pylab as plt

model = pgmpy.models.BayesianModel([('Burglary', 'Alarm'),

                                    ('Earthquake', 'Alarm'),

                                    ('Alarm', 'JohnCalls'),

                                    ('Alarm', 'MaryCalls')])

cpd\_burglary = pgmpy.factors.discrete.TabularCPD('Burglary', 2, [[0.001], [0.999]])

cpd\_earthquake = pgmpy.factors.discrete.TabularCPD('Earthquake', 2, [[0.002], [0.998]])

cpd\_alarm = pgmpy.factors.discrete.TabularCPD('Alarm', 2, [[0.95, 0.94, 0.29, 0.001],

                                                           [0.05, 0.06, 0.71, 0.999]],

                                              evidence=['Burglary', 'Earthquake'],

                                              evidence\_card=[2, 2])

cpd\_john = pgmpy.factors.discrete.TabularCPD('JohnCalls', 2, [[0.90, 0.05],

                                                           [0.10, 0.95]],

                                              evidence=['Alarm'],

                                              evidence\_card=[2])

cpd\_mary = pgmpy.factors.discrete.TabularCPD('MaryCalls', 2, [[0.70, 0.01],

                                                           [0.30, 0.99]],

                                              evidence=['Alarm'],

                                              evidence\_card=[2])

model.add\_cpds(cpd\_burglary, cpd\_earthquake, cpd\_alarm, cpd\_john, cpd\_mary)

model.check\_model()

print('Probability distribution, P(Burglary)')

print(cpd\_burglary)

print()

print('Probability distribution, P(Earthquake)')

print(cpd\_earthquake)

print()

print('Joint probability distribution, P(Alarm | Burglary, Earthquake)')

print(cpd\_alarm)

print()

print('Joint probability distribution, P(JohnCalls | Alarm)')

print(cpd\_john)

print()

print('Joint probability distribution, P(MaryCalls | Alarm)')

print(cpd\_mary)

print()

output:

+-------------+-------+

| Burglary(0) | 0.001 |

+-------------+-------+

| Burglary(1) | 0.999 |

+-------------+-------+

Probability distribution, P(Earthquake)

+---------------+-------+

| Earthquake(0) | 0.002 |

+---------------+-------+

| Earthquake(1) | 0.998 |

+---------------+-------+

Joint probability distribution, P(Alarm | Burglary, Earthquake)

+------------+---------------+---------------+---------------+---------------+

| Burglary | Burglary(0) | Burglary(0) | Burglary(1) | Burglary(1) |

+------------+---------------+---------------+---------------+---------------+

| Earthquake | Earthquake(0) | Earthquake(1) | Earthquake(0) | Earthquake(1) |

+------------+---------------+---------------+---------------+---------------+

| Alarm(0) | 0.95 | 0.94 | 0.29 | 0.001 |

+------------+---------------+---------------+---------------+---------------+

| Alarm(1) | 0.05 | 0.06 | 0.71 | 0.999 |

+------------+---------------+---------------+---------------+---------------+

Joint probability distribution, P(JohnCalls | Alarm)

+--------------+----------+----------+

| Alarm | Alarm(0) | Alarm(1) |

+--------------+----------+----------+

| JohnCalls(0) | 0.9 | 0.05 |

+--------------+----------+----------+

| JohnCalls(1) | 0.1 | 0.95 |

+--------------+----------+----------+

Joint probability distribution, P(MaryCalls | Alarm)

+--------------+----------+----------+

| Alarm | Alarm(0) | Alarm(1) |

+--------------+----------+----------+

| MaryCalls(0) | 0.7 | 0.01 |

+--------------+----------+----------+

| MaryCalls(1) | 0.3 | 0.99 |

+--------------+----------+----------+