K. J. Somaiya College of Engineering, Mumbai-77

Batch: A-4	**Roll No.:** 16010122151
Experiment No. 08	
Group No :- 5	
Signature of the Staff In-charge with date	

Title: Implementation of Bayesian networks

Objective: Understand inferences in uncertain scenarios

Expected Outcome of Experiment:

Course Outcome	After successful completion of the course students should be able to
CO 3	Represent and formulate the knowledge to solve the problems using various reasoning techniques

Books/ Journals/ Websites referred:

1. **"Artificial Intelligence: a Modern Approach" by Russel and Norving, Pearson education Publications**
2. **"Artificial Intelligence" By Rich and knight, Tata Mcgraw Hill Publications**
3. https://machinelearningmastery.com/introduction-to-bayesian-belief-networks/, last retried on April 02,2025
4. https://towardsdatascience.com/introduction-to-bayesian-belief-networks-c012e3f59f 1b , last retried on April 02,2025

Pre Lab/ Prior Concepts: Conditional Probability theory, Probability theory

Historical Profile: - Uncertainty is an inherent challenge in artificial intelligence (AI), arising from incomplete, noisy, or ambiguous information. In real-world scenarios, AI systems must make decisions despite lacking full knowledge of the environment. Addressing uncertainty is crucial for building robust and reliable AI models that can reason, learn, and adapt effectively. Bayesian networks provide a powerful probabilistic framework to represent and manage uncertainty by modelling dependencies between variables. They enable AI systems to make informed predictions, update beliefs with new evidence, and handle complex decision-making under uncertainty.

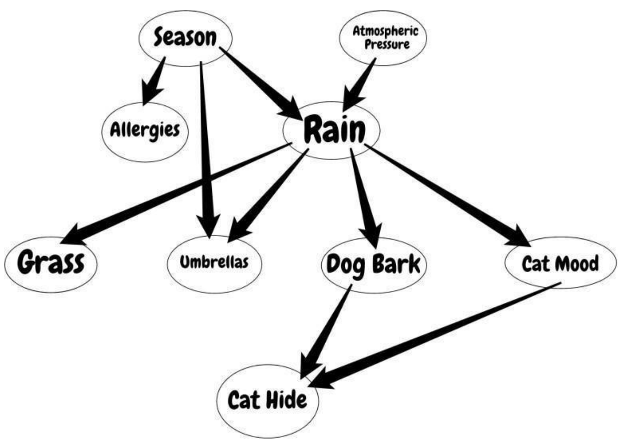
New Concepts to be learned: Uncertainty, reasoning with uncertain information, Bayesian network topology

Bayesian networks:

A Bayesian network (also known as a Bayes network, belief network, or decision network) is a probabilistic graphical model that represents a set of variables and their conditional dependencies via a directed acyclic graph (DAG). Bayesian networks are ideal for taking an event that occurred and predicting the likelihood that any one of several possible known causes was the contributing factor. For example, a Bayesian network could represent the probabilistic relationships between diseases and symptoms. Given symptoms, the network can be used to compute the probabilities of the presence of various diseases.

Efficient algorithms can perform inference and learning in Bayesian networks. Bayesian networks that model sequences of variables (e.g. speech signals or protein sequences) are called dynamic Bayesian networks. Generalizations of Bayesian networks that can represent and solve decision problems under uncertainty are called influence diagrams.

Bayesian Belief Network or Bayesian Network or Belief Network is a Probabilistic Graphical Model (PGM) that represents conditional dependencies between random variables through a Directed Acyclic Graph (DAG).



Bayesian Networks are applied in many fields. For example, disease diagnosis, optimized web search, spams filtering, gene regulatory networks, etc. And this list can be extended. The main objective of these networks is trying to understand the structure of causality relations. To clarify this, let's consider a disease diagnosis problem. With given symptoms and their resulting disease, we construct our Belief Network and when a new patient comes, we can infer which disease or diseases may have the new patient by providing probabilities for each disease. Similarly, these causality relations can be constructed for other problems and inference techniques can be applied

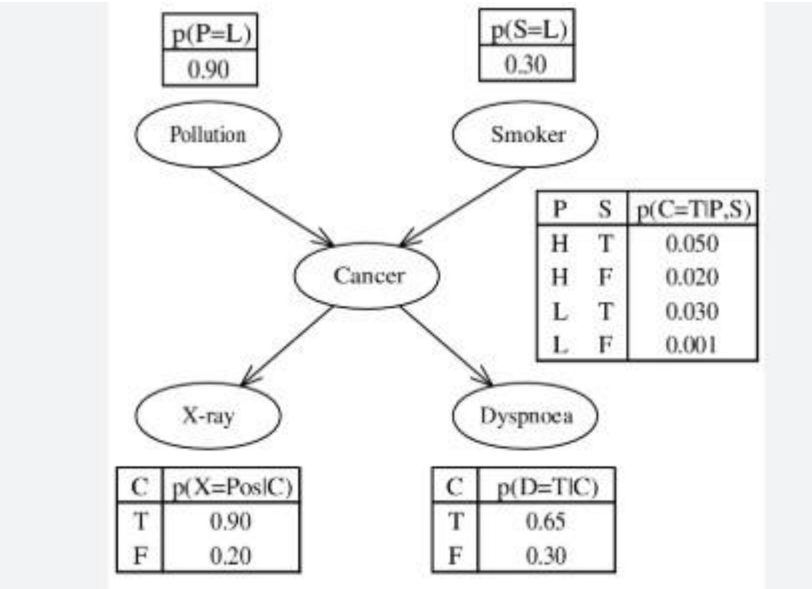
$$P(X|Y) = \frac{P(X,Y)}{P(Y)}$$

to interesting results.

As you would understand from the formula, to be able to calculate the joint distribution we need to have conditional probabilities indicated by the network. But further that if we have the joint distribution, then we can start to ask interesting questions. For example, in the first example, we ask for the probability of "RAIN" if "SEASON" is "WINTER" and "DOG BARK" is "TRUE".

Work Description: For the given problem, define the network, calculate the probabilities and query the system.

Chosen Problem statement: Lung Cancer Prediction Using Bayesian Network



Implementation:

```
!pip install pgmpy
```

```
Downloading nvidia_cuda_runtime_cu12-12.4.127-py3-none-manylinux2014_x86_64.whl (883 kB)
   ---------------------------------------- 883.7/883.7 kB 32.3 MB/s eta 0:00:00
Downloading nvidia_cudnn_cu12-9.1.0.70-py3-none-manylinux2014_x86_64.whl (664.8 MB)
   ---------------------------------------- 664.8/664.8 MB 3.0 MB/s eta 0:00:00
Downloading nvidia_cufft_cu12-11.2.1.3-py3-none-manylinux2014_x86_64.whl (211.5 MB)
   ---------------------------------------- 211.5/211.5 MB 5.7 MB/s eta 0:00:00
Downloading nvidia_curand_cu12-10.3.5.147-py3-none-manylinux2014_x86_64.whl (56.3 MB)
   ---------------------------------------- 56.3/56.3 MB 11.0 MB/s eta 0:00:00
Downloading nvidia_cusolver_cu12-11.6.1.9-py3-none-manylinux2014_x86_64.whl (127.9 MB)
   ---------------------------------------- 127.9/127.9 MB 7.2 MB/s eta 0:00:00
Downloading nvidia_cusparse_cu12-12.3.1.170-py3-none-manylinux2014_x86_64.whl (207.5 MB)
   ---------------------------------------- 207.5/207.5 MB 6.0 MB/s eta 0:00:00
Downloading nvidia_nvjitlink_cu12-12.4.127-py3-none-manylinux2014_x86_64.whl (21.1 MB)
   ---------------------------------------- 21.1/21.1 MB 78.9 MB/s eta 0:00:00
Downloading pyro_api-0.1.2-py3-none-any.whl (11 kB)
Installing collected packages: pyro-api, nvidia-nvjitlink-cu12, nvidia-curand-cu12, nvidia-cufft-cu12, nvidia-cuda-runtime-cu12, nvidia-cuda-nvrtc-cu12, nvidia-
  Attempting uninstall: nvidia-nvjitlink-cu12
    Found existing installation: nvidia-nvjitlink-cu12 12.5.82
    Uninstalling nvidia-nvjitlink-cu12-12.5.82:
      Successfully uninstalled nvidia-nvjitlink-cu12-12.5.82
  Attempting uninstall: nvidia-curand-cu12
    Found existing installation: nvidia-curand-cu12 10.3.6.82
    Uninstalling nvidia-curand-cu12-10.3.6.82:
      Successfully uninstalled nvidia-curand-cu12-10.3.6.82
  Attempting uninstall: nvidia-cufft-cu12
    Found existing installation: nvidia-cufft-cu12 11.2.3.61
    Uninstalling nvidia-cufft-cu12-11.2.3.61:
```

```python
import numpy as np

from pgmpy.models import BayesianNetwork

from pgmpy.factors.discrete import TabularCPD

from pgmpy.inference import VariableElimination

from pgmpy.models import DiscreteBayesianNetwork


model = DiscreteBayesianNetwork([

    ('Pollution', 'Cancer'),

    ('Smoker', 'Cancer'),

    ('Cancer', 'Xray'),

    ('Cancer', 'Dyspnoea')

])

cpd_pollution = TabularCPD(variable='Pollution', variable_card=2, values=[[0.9], [0.1]])

cpd_smoker = TabularCPD(variable='Smoker', variable_card=2, values=[[0.3], [0.7]])

cpd_cancer = TabularCPD(

    variable='Cancer', variable_card=2,

    values=[

        [0.03, 0.05, 0.02, 0.001],

        [0.97, 0.95, 0.98, 0.999]
```

```
    ],
    evidence=['Pollution', 'Smoker'],
    evidence_card=[2, 2]
)

cpd_xray = TabularCPD(
    variable='Xray', variable_card=2,
    values=[[0.9, 0.2],
            [0.1, 0.8]],
    evidence=['Cancer'],
    evidence_card=[2]
)
cpd_dyspnoea = TabularCPD(
    variable='Dyspnoea', variable_card=2,
    values=[[0.65, 0.3],
            [0.35, 0.7]],
    evidence=['Cancer'],
    evidence_card=[2]
)
model.add_cpds(cpd_pollution, cpd_smoker, cpd_cancer, cpd_xray, cpd_dyspnoea)
assert model.check_model()
print("Bayesian Network Model is valid!")

inference = VariableElimination(model)
```

Link to your colab notebook if you have implemented your own problem:

https://colab.research.google.com/drive/136TTqjfMwzp2ifpXOJ2HrufjB8XhAbDT#

Query 1:

```
print("\nP(Cancer | Smoker=True):")

query1 = inference.query(variables=["Cancer"],
evidence={"Smoker":0})

print(query1)
```

Computations for query 1:

```
P(Cancer | Smoker=True):
+-----------+---------------+
| Cancer    |   phi(Cancer) |
+===========+===============+
| Cancer(0) |        0.0290 |
+-----------+---------------+
| Cancer(1) |        0.9710 |
+-----------+---------------+
```

Query 2:

```
print("\nP(Cancer | Pollution=Low, Smoker=True):")
query2 = inference.query(variables=["Cancer"],
evidence={"Pollution": 0, "Smoker": 0})
print(query2)
```

Computations for query 2:

```
P(Cancer | Pollution=Low, Smoker=True):
+-----------+---------------+
| Cancer    |   phi(Cancer) |
+===========+===============+
| Cancer(0) |        0.0300 |
+-----------+---------------+
| Cancer(1) |        0.9700 |
+-----------+---------------+
```

Query 3:

```
print("\nP(Smoker | Cancer=True):")
query3 = inference.query(variables=["Smoker"],
evidence={"Cancer": 0})
print(query3)
```

Computations for query 3:

```
P(Smoker | Cancer=True):
+-----------+---------------+
| Smoker    |   phi(Smoker) |
+===========+===============+
| Smoker(0) |        0.2160 |
+-----------+---------------+
| Smoker(1) |        0.7840 |
+-----------+---------------+
```

Query 4:

```
print("\nP(Cancer | Xray=Positive):")
query4 = inference.query(variables=["Cancer"], evidence={"Xray":
0})
print(query4)
```

Computations for query 4:

```
P(Cancer | Xray=Positive):
+-----------+---------------+
| Cancer    |   phi(Cancer) |
+===========+===============+
| Cancer(0) |        0.1588 |
+-----------+---------------+
| Cancer(1) |        0.8412 |
+-----------+---------------+
```

Query 5:

```
print("\nP(Dyspnoea | Cancer=True):")
query5 = inference.query(variables=["Dyspnoea"],
evidence={"Cancer": 0})
print(query5)
```

Computations for query 5:

```
P(Dyspnoea | Cancer=True):
+-------------+-----------------+
| Dyspnoea    |   phi(Dyspnoea) |
+=============+=================+
| Dyspnoea(0) |          0.6500 |
+-------------+-----------------+
| Dyspnoea(1) |          0.3500 |
+-------------+-----------------+
```

PostLab Questions:

1. **Which of the following best describes a Bayesian Network?**
a) A network of independent random variables
b) A graphical model representing conditional dependencies among variables
c) A deterministic rule-based AI model
d) A deep learning neural network

Answer: b) A graphical model representing conditional dependencies among variables

2. **In a Bayesian Network, what do the edges between nodes represent?**
a) Causal or probabilistic dependencies
b) Logical equivalence
c) Time-dependent transitions
d) Random connections

Answer: a) Causal or probabilistic dependencies

Descriptive Questions:

1. **Explain the significance of Bayesian Networks in AI. How do they help in decision-making under uncertainty?**

   Bayesian Networks are crucial in AI for modeling uncertainty and reasoning under incomplete information. They represent variables and their conditional dependencies through a directed acyclic graph (DAG). In decision-making, Bayesian Networks:

   - Model real-world uncertainty by capturing probabilistic relationships.
   - Facilitate inference: Given evidence, they compute the posterior probabilities of unknown variables.
   - Support decision-making by offering probabilistic reasoning, e.g., estimating the likelihood of diseases, failures, or outcomes.
   - Enable explainability: Their graphical structure allows easy interpretation of how different variables influence each other.

2. **What are the main components of a Bayesian Network? Explain each briefly.**

   Nodes (Variables):
   Each node represents a random variable which could be discrete or continuous (e.g.,

Cancer, Smoker).

Edges (Directed Links):
Directed edges represent conditional dependencies or probabilistic relationships between variables (e.g., Pollution → Cancer).

Conditional Probability Tables (CPTs):
Each node has a CPT that quantifies the effect of the parent nodes. It shows the probability of a node given the values of its parents.

Directed Acyclic Graph (DAG):
The structure of the network, where nodes are connected with no cycles, ensuring logical consistency and efficient computation.

3. **Suppose you have a Bayesian Network with three variables: Disease, Test Result, and Symptoms. Explain how you would use conditional probabilities to determine the likelihood that a patient has the disease given a positive test result.**

To determine the probability that a patient has the disease given a positive test result (and possibly symptoms), follow these steps:

1. **Identify Variables:**
   - **D** = Disease (True/False)
   - **T** = Test Result (Positive/Negative)
   - **S** = Symptoms (Present/Absent)
2. **Use Bayes' Theorem:**

$$P(D = True|T = Positive) = \frac{P(T = Positive|D = True) \cdot P(D = True)}{P(T = Positive)}$$

3. **Expand Denominator:**

$$P(T = Positive) = P(T = Positive|D = True) \cdot P(D = True) + P(T = Positive|D = False) \cdot P(D = False)$$

4. **Include Symptoms if Available:** Use joint probabilities:

$$P(D = True|T = Positive, S = Present)$$

Which would involve applying the chain rule and CPTs in the Bayesian Network.

5. **Inference:** Using the CPTs and observed evidence (like T=Positive, S=Present), compute the posterior probability to make a data-driven medical diagnosis.

Example:

Let's assume test result is **positive**:

P(D=True|T=Positive) Apply

Bayes' Theorem:

$$P(D = True|T = Positive) = \frac{P(T = Positive|D = True) \cdot P(D = True)}{P(T = Positive)}$$

Where:

$$P(T = Positive) = P(T = Positive|D = True) \cdot P(D = True) + P(T = Positive|D = False) \cdot P(D = False)$$

$$= (0.95)(0.01) + (0.05)(0.99) = 0.0095 + 0.0495 = 0.059$$

$$P(D = True|T = Positive) = \frac{0.0095}{0.059} \approx 0.161$$

So, even though the test is positive, the actual chance of having the disease is only **16.1%** due to the **low prior probability** of disease.

Conclusion: In the context of lung cancer prediction, a Bayesian Network can effectively integrate various risk factors like pollution, smoking habits, and symptoms to infer the likelihood of a patient having cancer. This aids healthcare professionals in making informed, data-driven decisions, thereby improving diagnostic accuracy and patient outcomes.