**Batch: A-3 Roll No.: 16010122104**

**Experiment / assignment / tutorial No. 8**

|  |
| --- |
| **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](https://towardsdatascience.com/introduction-to-bayesian-belief-networks-c012e3f59f1b) , 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

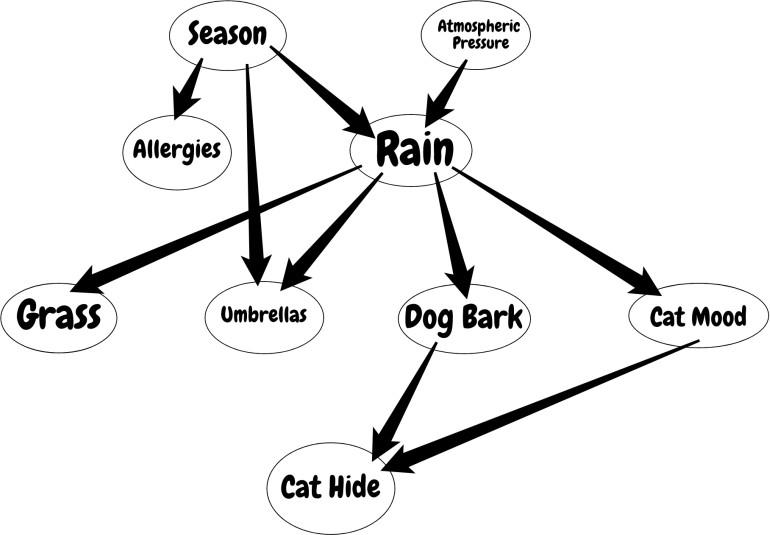
**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

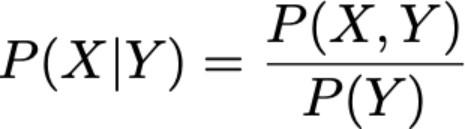
# Bayesian networks:

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 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:**

**To understand and predict how various factors influence a tourist’s visit satisfaction in a city (like Mumbai), especially focusing on how things like season, weather, major events, crowds, and hotel prices are related.**

**Implementation:**

from pgmpy.models import DiscreteBayesianNetwork

from pgmpy.factors.discrete import TabularCPD

from pgmpy.inference import VariableElimination

model = DiscreteBayesianNetwork([

    ('Season', 'Weather'),

    ('Season', 'Major Event'),

    ('Weather', 'Crowds'),

    ('Major Event', 'Crowds'),

    ('Crowds', 'Hotel Price'),

    ('Major Event', 'Hotel Price'),

    ('Weather', 'Visit Satisfaction'),

    ('Crowds', 'Visit Satisfaction'),

    ('Hotel Price', 'Visit Satisfaction')

])

cpd\_season = TabularCPD(

    variable='Season', variable\_card=2,

    values=[[0.5], [0.5]],

    state\_names={'Season': ['Summer', 'Winter']}

)

cpd\_weather = TabularCPD(

    variable='Weather', variable\_card=2,

    values=[[0.8, 0.3],   # Sunny: Summer, Winter

            [0.2, 0.7]],  # Rainy: Summer, Winter

    evidence=['Season'], evidence\_card=[2],

    state\_names={'Weather': ['Sunny', 'Rainy'], 'Season': ['Summer', 'Winter']}

)

cpd\_major\_event = TabularCPD(

    variable='Major Event', variable\_card=2,

    values=[[0.4, 0.1],   # Yes: Summer, Winter

            [0.6, 0.9]],  # No: Summer, Winter

    evidence=['Season'], evidence\_card=[2],

    state\_names={'Major Event': ['Yes', 'No'], 'Season': ['Summer', 'Winter']}

)

cpd\_crowds = TabularCPD(

    variable='Crowds', variable\_card=2,

    values=[

        [0.95, 0.6, 0.7, 0.2],  # High: [Sunny,Yes], [Sunny,No], [Rainy,Yes], [Rainy,No]

        [0.05, 0.4, 0.3, 0.8]   # Low:  [Sunny,Yes], [Sunny,No], [Rainy,Yes], [Rainy,No]

    ],

    evidence=['Weather', 'Major Event'], evidence\_card=[2, 2],

    state\_names={

        'Crowds': ['High', 'Low'],

        'Weather': ['Sunny', 'Rainy'],

        'Major Event': ['Yes', 'No']

    }

)

cpd\_hotel\_price = TabularCPD(

    variable='Hotel Price', variable\_card=2,

    values=[

        [0.9, 0.5, 0.6, 0.1],  # Expensive: [High,Yes], [High,No], [Low,Yes], [Low,No]

        [0.1, 0.5, 0.4, 0.9]   # Affordable: [High,Yes], [High,No], [Low,Yes], [Low,No]

    ],

    evidence=['Crowds', 'Major Event'], evidence\_card=[2, 2],

    state\_names={

        'Hotel Price': ['Expensive', 'Affordable'],

        'Crowds': ['High', 'Low'],

        'Major Event': ['Yes', 'No']

    }

)

cpd\_visit\_satisfaction = TabularCPD(

    variable='Visit Satisfaction', variable\_card=2,

    values=[

        # High Satisfaction

        [0.7, 0.6, 0.8, 0.95, 0.5, 0.4, 0.6, 0.3],

        # Low Satisfaction

        [0.3, 0.4, 0.2, 0.05, 0.5, 0.6, 0.4, 0.7]

    ],

    evidence=['Weather', 'Crowds', 'Hotel Price'], evidence\_card=[2, 2, 2],

    state\_names={

        'Visit Satisfaction': ['High', 'Low'],

        'Weather': ['Sunny', 'Rainy'],

        'Crowds': ['High', 'Low'],

        'Hotel Price': ['Expensive', 'Affordable']

    }

)

model.add\_cpds(

    cpd\_season, cpd\_weather, cpd\_major\_event,

    cpd\_crowds, cpd\_hotel\_price, cpd\_visit\_satisfaction

)

assert model.check\_model()

inference = VariableElimination(model)

# Example queries

# Query 1: P(Visit Satisfaction=High | Season=Summer, Major Event=No)

query1 = inference.query(

    variables=['Visit Satisfaction'],

    evidence={'Season': 'Summer', 'Major Event': 'No'}

)

# Query 2: P(Hotel Price=Expensive | Season=Winter, Major Event=Yes)

query2 = inference.query(

    variables=['Hotel Price'],

    evidence={'Season': 'Winter', 'Major Event': 'Yes'}

)

# Query 3: P(Crowds=High | Weather=Sunny, Major Event=Yes)

query3 = inference.query(

    variables=['Crowds'],

    evidence={'Weather': 'Sunny', 'Major Event': 'Yes'}

)

# Query 4: P(Weather=Sunny | Season=Winter)

query4 = inference.query(

    variables=['Weather'],

    evidence={'Season': 'Winter'}

)

# Query 5: P(Visit Satisfaction=High | Hotel Price=Affordable, Crowds=Low)

query5 = inference.query(

    variables=['Visit Satisfaction'],

    evidence={'Hotel Price': 'Affordable', 'Crowds': 'Low'}

)

# Print results

print("Query 1:", query1)

print("Query 2:", query2)

print("Query 3:", query3)

print("Query 4:", query4)

print("Query 5:", query5)

**Query 1: +--------------------------+---------------------------+**

**| Visit Satisfaction | phi(Visit Satisfaction) |**

**+==========================+===========================+**

**| Visit Satisfaction(High) | 0.6820 |**

**+--------------------------+---------------------------+**

**| Visit Satisfaction(Low) | 0.3180 |**

**+--------------------------+---------------------------+**

**Query 2: +-------------------------+--------------------+**

**| Hotel Price | phi(Hotel Price) |**

**+=========================+====================+**

**| Hotel Price(Expensive) | 0.8325 |**

**+-------------------------+--------------------+**

**| Hotel Price(Affordable) | 0.1675 |**

**+-------------------------+--------------------+**

**Query 3: +--------------+---------------+**

**| Crowds | phi(Crowds) |**

**+==============+===============+**

**| Crowds(High) | 0.9500 |**

**+--------------+---------------+**

**| Crowds(Low) | 0.0500 |**

**+--------------+---------------+**

**Query 4: +----------------+----------------+**

**| Weather | phi(Weather) |**

**+================+================+**

**| Weather(Sunny) | 0.3000 |**

**+----------------+----------------+**

**| Weather(Rainy) | 0.7000 |**

**+----------------+----------------+**

**Query 5: +--------------------------+---------------------------+**

**| Visit Satisfaction | phi(Visit Satisfaction) |**

**+==========================+===========================+**

**| Visit Satisfaction(High) | 0.5156 |**

**+--------------------------+---------------------------+**

**| Visit Satisfaction(Low) | 0.4844 |**

**+--------------------------+---------------------------+**

**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:**

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:**

**Descriptive Questions:**

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

**Ans:**

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

**Ans:**

1. **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.**

**Ans:**

**Conclusion:**