Exp-1

Name: Vaibhav Boudh
Division: D20B
Roll no: 06

Aim: To Implement Inferencing with Bayesian Network in Python

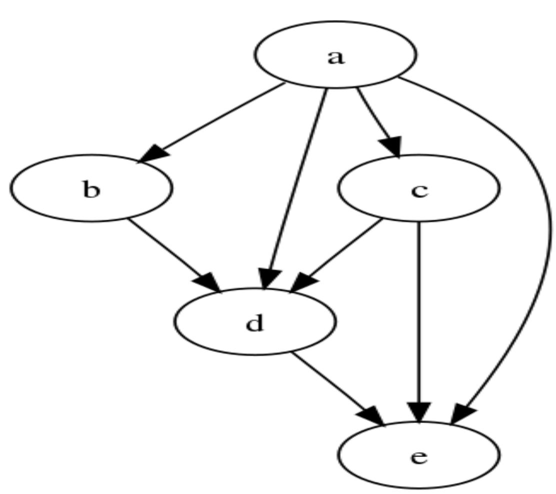
Theory:
Bayesian Belief Networks (BBNs) are a key tool in Artificial Intelligence for representing and reasoning under uncertainty using probabilistic graphical models. A BBN consists of a **Directed Acyclic Graph (DAG)** where each node represents a random variable and each edge represents conditional dependency.
A Bayesian Network enables inferencing by using **Conditional Probability Tables (CPTs)** and algorithms like **Variable Elimination** to compute the probability of a variable given known evidence.

Applications of Bayesian Belief Networks:

1. **Medical Diagnosis** – Predict diseases based on symptoms
2. **Fault Diagnosis** – Detect faults in machines or IT systems
3. **Decision Support Systems** – Aid business/environmental decisions
4. **AI & Robotics** – Path planning and decision-making in uncertainty
5. **Speech & Image Recognition** – NLP and computer vision tasks
6. **Bioinformatics** – Gene expression analysis
7. **Finance** – Risk assessment and portfolio management
8. **Environmental Science** – Ecosystem modeling
9. **Forensics** – Evidence-based scenario evaluation
10. **Education** – Adaptive learning systems

Structure of Bayesian Network:

- **DAG**: Represents variables and their dependencies



- **CPT**: Specifies the probability distribution for each node given its parents

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

- **Joint Probability Distribution**:

$$P(X_1, X_2, \ldots, X_n) = \prod_{i=1}^{n} P(X_i \mid \text{Parents}(X_i))$$

Test Cases:

- **Example 1:** P(Buy House,Marry∣Salaried=1,Handsome=0)
- **Example 2:** P(Buy House,Marry∣Salaried=0,Handsome=0)
- **Example 3:** P(Buy House,Marry∣Salaried=0,Handsome=1)

Code Implementation (Python):

```python
from pgmpy.models import DiscreteBayesianNetwork
from pgmpy.factors.discrete import TabularCPD
from pgmpy.inference import VariableElimination
# Define the Bayesian Network structure
model = DiscreteBayesianNetwork([
    ('Salaried', 'Marry'),
    ('Handsome', 'Marry'),
    ('Marry', 'Buy House')
])
# Define the CPDs
cpd_salaried = TabularCPD(variable='Salaried', variable_card=2, values=[[0.2], [0.8]])
cpd_handsome = TabularCPD(variable='Handsome', variable_card=2, values=[[0.35], [0.65]])
cpd_marry = TabularCPD(variable='Marry', variable_card=2,
                       values=[[0.05, 0.32, 0.47, 1.0],
                               [0.95, 0.68, 0.53, 0.0]],
                       evidence=['Salaried', 'Handsome'],
                       evidence_card=[2, 2])
cpd_buy_house = TabularCPD(variable='Buy House', variable_card=2,
                           values=[[0.08, 0.79],
                                   [0.92, 0.21]],
                           evidence=['Marry'],
                           evidence_card=[2])
# Add CPDs to the model
model.add_cpds(cpd_salaried, cpd_handsome, cpd_marry, cpd_buy_house)
assert model.check_model()
# Inference
inference = VariableElimination(model)
# Query example
query = inference.query(variables=['Buy House', 'Marry'],
                        evidence={'Salaried': 1, 'Handsome': 0},
                        joint=True)
print(query)
```

Output:

```
Test Case 1: Salaried=1, Handsome=0
+--------------+----------+-----------------------+
| Buy House    | Marry    |   phi(Buy House,Marry) |
+==============+==========+=======================+
| Buy House(0) | Marry(0) |                0.0376 |
+--------------+----------+-----------------------+
| Buy House(0) | Marry(1) |                0.4187 |
+--------------+----------+-----------------------+
| Buy House(1) | Marry(0) |                0.4324 |
+--------------+----------+-----------------------+
| Buy House(1) | Marry(1) |                0.1113 |
+--------------+----------+-----------------------+
```

```
Test Case 2: Salaried=0, Handsome=0
+--------------+----------+-----------------------+
| Buy House    | Marry    |   phi(Buy House,Marry) |
+==============+==========+=======================+
| Buy House(0) | Marry(0) |                0.0040 |
+--------------+----------+-----------------------+
| Buy House(0) | Marry(1) |                0.7505 |
+--------------+----------+-----------------------+
| Buy House(1) | Marry(0) |                0.0460 |
+--------------+----------+-----------------------+
| Buy House(1) | Marry(1) |                0.1995 |
+--------------+----------+-----------------------+
```

```
Test Case 3: Salaried=0, Handsome=1
+--------------+----------+-----------------------+
| Buy House    | Marry    |   phi(Buy House,Marry) |
+==============+==========+=======================+
| Buy House(0) | Marry(0) |                0.0256 |
+--------------+----------+-----------------------+
| Buy House(0) | Marry(1) |                0.5372 |
+--------------+----------+-----------------------+
| Buy House(1) | Marry(0) |                0.2944 |
+--------------+----------+-----------------------+
| Buy House(1) | Marry(1) |                0.1428 |
+--------------+----------+-----------------------+
```

Conclusion:

We have successfully implemented **inferencing with a Bayesian Network** in Python using the pgmpy library and verified its outputs on different inputs using variable elimination.