Machine Learning (ML)

Name : Prekha A. Patel
Sapid : 60004210126
Branch : Computer Engineering
Div : C2, Batch : 1

Experiment no. 9

Aim : To implement Bayesian Belief Network.

Theory : Bayesian classification is based on Bayesian Theorem. Bayesian classifiers are the statistical classifiers. They can predict class membership possibilities such as the probability that a given tuple belongs to a particular class.

Bayesian Belief Network specify joint conditional probability distributions. we can use a trained Bayesian Network for classification.

There are two components that define a Bayesian Belief Network. Directed acyclic graph. A set of conditional probability tables Directed Acyclic Graph. Each node in a directed acyclic graph represents a random variable. These variables may be discrete and continuous valued. These variables may correspond to actual attribute given in the data.

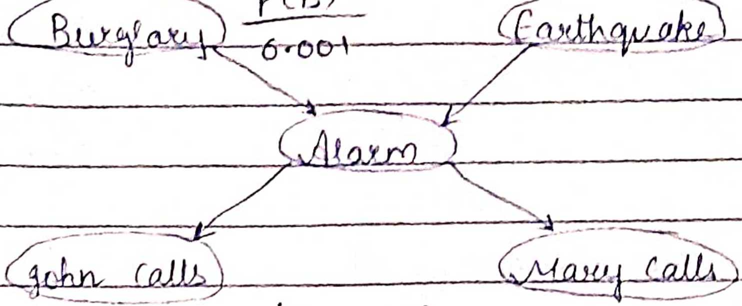
Directed acyclic graph

- Node = Random variables

Burglary, Earthquake, Alarm, Mary calls, John calls

- links : direct (casual) dependencies between variables.

The chance of Alarm is induced by Earthquake, The chance of John calling is affected by Alarm.



P(B)
0.001

P(E)
0.002

A	P(J/A)
T	0.90
F	0.05

A	P(M/A)
T	0.70
F	0.01

B	E	P(A|B, E)
T	T	0.95
T	F	0.94
F	T	0.29
F	F	0.001

Q1] What is the probability that the alarm has sounded but neither a burglary nor an earthquake has occured and both John and Merry call?

→

$$P(j \wedge m \wedge a \wedge \neg b \wedge \neg e) = P(j|a)\,P(m|a)\,P(a|\neg b, \neg e)\,P(\neg b)\,P(\neg e)$$

$$= 0.90 \times 0.70 \times 0.001 \times 0.999 \times 0.998$$

$$= 0.00062$$

Q2] What is the probability that John calls?

→

$$P(j) = P(j|a)\,P(a) + P(j|\neg a)\,P(\neg a)$$

$$= P(j|a)\{P(a|b,e) * P(b,e) + P(a|\neg b, e) \times P(\neg b, e) + P(a|b,\neg e) \times P(b,\neg e) + P(a|\neg b, \neg e) \times P(\neg b, \neg e)\}$$

$$+ P(P_j|\neg a)\{P(\neg a|b,e) \times P(b,e) + P(\neg a|\neg b, e) \times P(\neg b, e) + P(\neg a b, \neg e) \times P(b, \neg e) + P(\neg a|\neg b, \neg e) \times P(\neg b, \neg e)\}$$

$$= 0.90 \times 0.00252 + 0.05 \times 0.9974$$

$$= 0.0521$$

SDS Page No.
Date

3] Probability that there is burglary given that john and mary calls.

$$P(\neg b \mid j, m) = \alpha P(\neg b) \sum_a P(j \mid a) P(m \mid a) \sum_e P(a \mid \neg b, e) P(e)$$

$$= \alpha P(\neg b) \sum_a P(j \mid a) P(m \mid a) \{ P(a \mid \neg b, e) P(e) + P(a \mid \neg b, \neg e) P(\neg e) \}$$

$$= \alpha P(\neg b) [P(j \mid a) P(m \mid a) \{ P(a \mid \neg b, e) P(e) + P(a \mid \neg b, \neg e) P(\neg e) \} + P(j \mid \neg a) P(m \mid \neg a) \{ P(\neg a \mid \neg b, e) P(e) + P(\neg a \mid \neg b, \neg e) P(\neg e) \}]$$

$$= \alpha \times 0.999 \times (0.9 \times 0.7 \times (0.29 \times 0.002 + 0.001 \times 0.998) + 0.05 \times 0.01 \times (0.71 \times 0.002 + 0.999 \times 0.998))$$

$$= \alpha \times 0.0015$$

$$\therefore \alpha = \frac{1}{(P(b, j, m) + P(\neg b, j, m))}$$

$$\alpha = \frac{1}{(0.00059 + 0.0015)}$$

$$\alpha = 478.5$$

$$P(b \mid j, m) = \alpha \times P(b, j, m)$$
$$= 478.5 \times 0.00059$$
$$= 0.28$$

$$P(\neg b \mid j, m) = \alpha \times P(\neg b, j, m)$$
$$= 478.5 \times 0.0015$$
$$= 0.72$$

P.T.O.

Conclusion :-

The output provides the joint probability of John calls, Mary calls & alarm given no burglary and no Earthquake, in the 1st query in the code.

In Query 2, in code, the output gives the likelihood of John making a call regardless of other events in network.

In Query 3, the Output offers insights into the likelihood of a burglary occurring when both John & Mary have called.

NAME:-PREKSHA ASHOK PATEL
SAPID:-60004210126
BRANCH:-COMPUTER ENGINEERING
DIV:-C2 ; BATCH:- 1

MACHINE LEARNING
EXPERIMENT NO. 09

CODE AND OUTPUT:-

```
[39] !pip install sorobn

     Requirement already satisfied: sorobn in /usr/local/lib/python3.10/dist-packages (0.3.0)
     Requirement already satisfied: numpy<2.0.0,>=1.24.2 in /usr/local/lib/python3.10/dist-packages (from sorobn) (1.25.2)
     Requirement already satisfied: pandas<2.0.0,>=1.5.3 in /usr/local/lib/python3.10/dist-packages (from sorobn) (1.5.3)
     Requirement already satisfied: vose<0.0.2,>=0.0.1 in /usr/local/lib/python3.10/dist-packages (from sorobn) (0.0.1)
     Requirement already satisfied: python-dateutil>=2.8.1 in /usr/local/lib/python3.10/dist-packages (from pandas<2.0.0,>=1.5.3->sorobn) (2.8.2)
     Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-packages (from pandas<2.0.0,>=1.5.3->sorobn) (2023.4)
     Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.8.1->pandas<2.0.0,>=1.5.3->sorobn) (1.16.0)

[40] import sorobn as soro
     import pandas as pd
     bayesian = soro.BayesNet(
          ('Burglary', 'Alarm'),
        ('Earthquake', 'Alarm'),
          ('Alarm', 'John calls'),
         ('Alarm', 'Mary calls'),
          seed=42,
      )
     bayesian.P['Burglary'] = pd.Series({False: .999, True: .001})
     bayesian.P['Earthquake'] = pd.Series({False: .998, True: .002})
```

```
[ ] # Probability(Alarm | Burglary, Earthquake)
    bayesian.P['Alarm'] = pd.Series({
        (True, True, True): .95,
        (True, True, False): .05,

        (True, False, True): .94,
        (True, False, False): .06,

        (False, True, True): .29,
       (False, True, False): .71,

         (False, False, True): .001,
         (False, False, False): .999 })
    # Probability(John calls | Alarm)
    bayesian.P['John calls'] = pd.Series({
        (True, True): .9,
        (True, False): .1,
        (False, True): .05,
        (False, False): .95
        })
    # Probability(Mary calls | Alarm)
    bayesian.P['Mary calls'] = pd.Series({
        (True, True): .7,
        (True, False): .3,
        (False, True): .01,
         (False, False): .99
     })
```

```
[42] bayesian.prepare()
```

QUERY 1

```
bayesian.query('Alarm','John calls','Mary calls' ,event={ 'Burglary': False, 'Earthquake': False})
```

```
Alarm  John calls  Mary calls
False  False       False         0.939560
                   True          0.009491
       True        False         0.049451
                   True          0.000500
True   False       False         0.000030
                   True          0.000070
       True        False         0.000270
                   True          0.000630
Name: P(Alarm, John calls, Mary calls), dtype: float64
```

QUERY 2

```
[44] bayesian.query('John calls',event={'Alarm':True,'Alarm':False})
```

```
John calls
False    0.95
True     0.05
Name: P(John calls), dtype: float64
```

QUERY 3

```
bayesian.query('Burglary', event={'Mary calls': True, 'John calls': True})
```

```
Burglary
False    0.715828
True     0.284172
Name: P(Burglary), dtype: float64
```

```
[46] print("Probabilities for B, E, A, John calls, and Mary calls:")
     print("-" * 50)
     print("| Variable | Value | Probability |")
     print("-" * 50)

     for variable in ["Burglary", "Earthquake", "Alarm", "John calls", "Mary calls"]:
         for value in bayesian.P[variable].index:
             probability = bayesian.P[variable][value]

             print(f"| {variable} | {value} | {probability} |")


     print("-" * 50)
```

```
Probabilities for B, E, A, John calls, and Mary calls:
--------------------------------------------------
| Variable | Value | Probability |
--------------------------------------------------
| Burglary | False | 0.999 |
| Burglary | True | 0.001 |
| Earthquake | False | 0.998 |
| Earthquake | True | 0.002 |
| Alarm | (False, False, False) | 0.999 |
| Alarm | (False, False, True) | 0.001 |
| Alarm | (False, True, False) | 0.71 |
| Alarm | (False, True, True) | 0.29 |
| Alarm | (True, False, False) | 0.06 |
| Alarm | (True, False, True) | 0.94 |
| Alarm | (True, True, False) | 0.05 |
| Alarm | (True, True, True) | 0.95 |
| John calls | (False, False) | 0.95 |
| John calls | (False, True) | 0.05 |
| John calls | (True, False) | 0.1 |
| John calls | (True, True) | 0.9 |
| Mary calls | (False, False) | 0.99 |
| Mary calls | (False, True) | 0.01 |
| Mary calls | (True, False) | 0.3 |
| Mary calls | (True, True) | 0.7 |
--------------------------------------------------
```