Experiment 9

Aim: Implementation of Bayes Belief Network

Objective: To study about how to use Bayes Belief Network in reasoning process.

Theory:

Bayesian Belief Network or Bayesian Network or Belief Network is a Probabilistic GraphicalModel (PGM) that represents conditional dependencies between random variables through a Directed Acyclic Graph (DAG). Bayesian Networks are applied in many fields. The main objective of these networks is trying to understand the structure of causality relations.

For example, disease diagnosis, optimized web search, spam filtering, gene regulatory networks, etc.

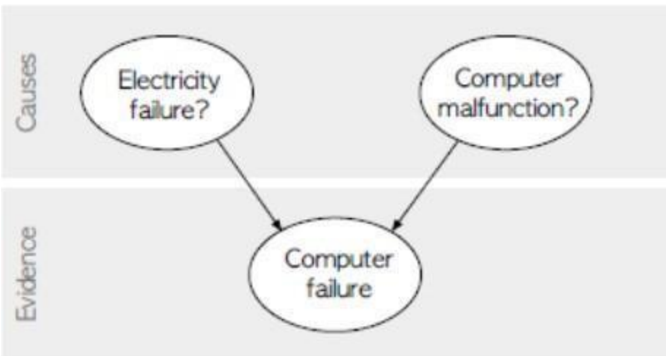
Bayesian Belief Network is a graphical representation of different probabilistic relationships among random variables in a particular set. It is a classifier with no dependency on attributes
i.e it is condition independent. Due to its feature of joint probability, the probability in BayesianBelief Network is derived, based on a condition — P(attribute/parent) i.e probability of an attribute, true over parent attribute.

A Bayesian network represents the causal probabilistic relationship among a set of random variables, their conditional dependences, and it provides a compact representationof a joint probability distribution. It consists of two major parts: a directed acyclic graph and a set of conditional probability distributions. The directed acyclic graph is a set of random variablesrepresented by nodes. For health measurement, a node may be a health domain, and the statesof the node would be the possible responses to that domain. If there exists a causal probabilisticdependence between two random variables in the graph, the corresponding two nodes are connected by a directed edge, while the directed edge from a node A to a node B indicatesthat the random variable A causes the random variable B. Since the directed edges represent a static causal probabilistic dependence, cycles are not allowed in the graph. A conditional probability distribution is defined for each node in the graph. In other words, the conditional probability distribution of a node (random variable) is defined for every possible outcome of the preceding causal node(s).

Example 1:

Suppose we attempt to turn on our computer, but the computer does not start(observation/evidence). We would like to know which of the possible causes of

computer failure is more likely. In this simplified illustration, we assume only two possible causes of this misfortune: electricity failure and computer malfunction. The corresponding directed acyclic graph is depicted in figure.



The two causes in this banal example are assumed to be independent (there is no edge betweenthe two causal nodes), but this assumption is not necessary in general. Unless there is a cycle in the graph, Bayesian networks are able to capture as many causal relations as it is necessary to credibly describe the real-life situation. Since a directed acyclic graph represents a hierarchical arrangement, it is unequivocal to use terms such as parent, child, ancestor, or descendant for certain node.

In figure, both electricity failure and computer malfunction are ancestors and parents of computer failure; analogically computer failure is a descendant and a child of both electricity failure and computer malfunction.

The goal is to calculate the posterior conditional probability distribution of each of the possibleunobserved causes given the observed evidence, i.e. P [Cause | Evidence]. However, inpractice we are often able to obtain only the converse conditional probability distribution of observing evidence given the cause, P [Evidence j Cause]. The whole concept of Bayesian networks is built on Bayes theorem, which helps us to express the conditional probability distribution of cause given the observed evidence using the converse conditional probability ofobserving evidence given the cause:

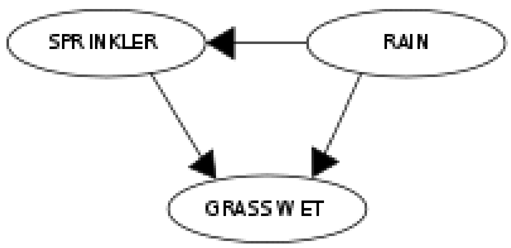
$$\mathrm{P}\,[\mathrm{Cause} \mid \mathrm{Evidence}] = \mathrm{P}\,[\mathrm{Evidence} \mid \mathrm{Cause}] \cdot \frac{\mathrm{P}\,[\mathrm{Cause}]}{\mathrm{P}\,[\mathrm{Evidence}]}$$

Any node in a Bayesian network is always conditionally independent of its all non-

descendants given that node's parents. Hence, the joint probability distribution of all random variables in the graph factorizes into a series of conditional probability distributions of randomvariables given their parents. Therefore, we can build a full probability model by onlyspecifying the conditional probability distribution in every node

Example 2: A Bayesian network with conditional probability tables

	SPRINKLER	
RAIN	T	F
F	0.4	0.6
T	0.01	0.99



RAIN	
T	F
0.2	0.8

		GRASS WET	
SPRINKLER	RAIN	T	F
F	F	0.0	1.0
F	T	0.8	0.2
T	F	0.9	0.1
T	T	0.99	0.01

Code:

```
knowledge_base = {
  'rule1': {
    'condition': lambda person: person['age'] >= 18,
    'result': 'You are eligible for a loan.'
  },
  'rule2': {
    'condition': lambda person: person['income'] > 30000,
    'result': 'You meet the income requirement for a loan.'
  },
  'rule3': {
    'condition': lambda person: person['credit_score'] >= 650,
    'result': 'You have a good credit score.'
  },
  'rule4': {
    'condition': lambda person: person['employment_status'] == 'employed',
    'result': 'You have a stable job.'
  },
  'rule5': {
    'condition': lambda person: person['age'] <= 60,
    'result': 'Your age is within the acceptable range for a loan.'
  },
  'rule6': {
    'condition': lambda person: person['income'] > 50000,
    'result': 'Your income is above the threshold for a higher loan amount.'
  },
  'rule7': {
    'condition': lambda person: person['credit_score'] >= 750,
    'result': 'Your excellent credit score qualifies you for lower interest rates.'
  },
  'rule8': {
    'condition': lambda person: person['employment_status'] == 'self-employed',
    'result': 'Your self-employment status is considered for a loan application.'
  }
}

def evaluate_rules(person):
  results = []

  for rule_name in ['rule1', 'rule2', 'rule3', 'rule4']:
    if not knowledge_base[rule_name]['condition'](person):
      return results

    results.append(knowledge_base[rule_name]['result'])

  return results

age = int(input("Enter your age: "))
income = float(input("Enter your annual income: "))
credit_score = int(input("Enter your credit score: "))
employment_status = input("Enter your employment status (employed/unemployed/self-employed): ")
```

```
user_person = {
  'age': age,
  'income': income,
  'credit_score': credit_score,
  'employment_status': employment_status
}

eligibility_results = evaluate_rules(user_person)

if eligibility_results:
  print("Loan Eligibility Results:")
  for result in eligibility_results:
    print(result)
else:
  print("You are not eligible for a loan.")
```

Output:

```
Enter your age: 25
Enter your annual income: 40000
Enter your credit score: 720
Enter your employment status (employed/unemployed/self-employed): employed

Loan Eligibility Results:
You are eligible for a loan.
You meet the income requirement for a loan.
You have a good credit score.
You have a stable job.
```

Conclusion:

Thus, we have studied about how to use Bayes Belief Network in reasoning process. Bayesian Belief Networks are Probabilistic Graphical Models represented as Directed Acyclic Graphs that model conditional dependencies between random variables to understand causal structure.