KNOWLEDGE REPRESENTATION

ASSIGNMENT – 2

1. Logical Reasoning:

a) I have described the predicates for the given relations. Since the predicates are relational, we need to read them as

relation(a,b) => a is a relation of b

- $\forall_{x,y}$Grandchild(x,y)=>parent(y,p)∧parent(p,x)

X is a Grandchild of Y only if Y is parent of some ‘p’ and ‘p’ is parent of X.

- $\forall_{x,y}$Greatgrandparent(x,y)=>parent(x,p)∧Grandchild(y,p)

X is a Greatgrandparent of Y only if X is parent of some ‘p’ and Y is Grandchild of ‘p’.

- $\forall_{x,y}$Ancestor(x,y)=>parent(x,y)
  $\forall_{x,y}$Ancestor(x,y)=>parent(x,p)∧parent(p,y)
  $\forall_{x,y}$Ancestor(x,y)=>parent(x,p)∧parent(p,q)∧parent(q,y)
  …

X can be an ancestor of Y if X is parent of Y or X is recursively parent of Y.

Ancestor is "any person from whom one is descended. In law, the person from whom an estate has been inherited."

- Let me define an additional relation that helps us in describing further relations.
  $\forall_{x,y}$Sibling(x,y)=>parent(p,x)∧parent(p,y)∧not_equal(x,y)
  Sibling (x, y) => Sibling (y, x)
- $\forall_{x,y}$Brother(x,y)=>Sibling(x,y)∧Gender(x,‘Male’)

X is a brother of Y when both of them are having same parent and when X’s gender is male.

- $\forall_{x,y}$Sister(x,y)=>Sibling(x,y)∧Gender(x,‘Female’)

X is a sister of Y when both of them are having same parent and when X’s gender is female.

- $\forall_{x,y}$Daughter(x,y)⇒parent(y,x)∧Gender(x,'Female')

X is daughter of Y when Y is parent of X and X’s gender is female.

- $\forall_{x,y}$Son(x, y)⇒parent(y,x)∧Gender(x,'Male')

X is son of Y when Y is parent of X and X’s gender is male.

- $\forall_{x,y}$FirstCousin(x,y)⇒parent(p,x)∧parent(q,y)∧Sibling(p,q)

X is first cousin of Y when their parents are siblings.

- Let me describe a relation married which helps in describing the upcoming relations.
  $\forall_{x,y}$Married(x,y)=>Married(y,x)
- $\forall_{x,y}$BrotherInLaw(x,y)⇒Married(y,p)∧Sibling(x,p)∧Gender(x,'Male')

X is brother-in-law of Y when Y is married to some ‘p’ and X is sibling of ‘p’ and X’s gender is male.

- $\forall_{x,y}$SisterInLaw(x,y)⇒Married(y,p)∧Sibling(x,p)∧Gender(x,'Female')

X is sister-in-law of Y when Y is married to some ‘p’ and X is sibling of ‘p’ and X’s gender is female.

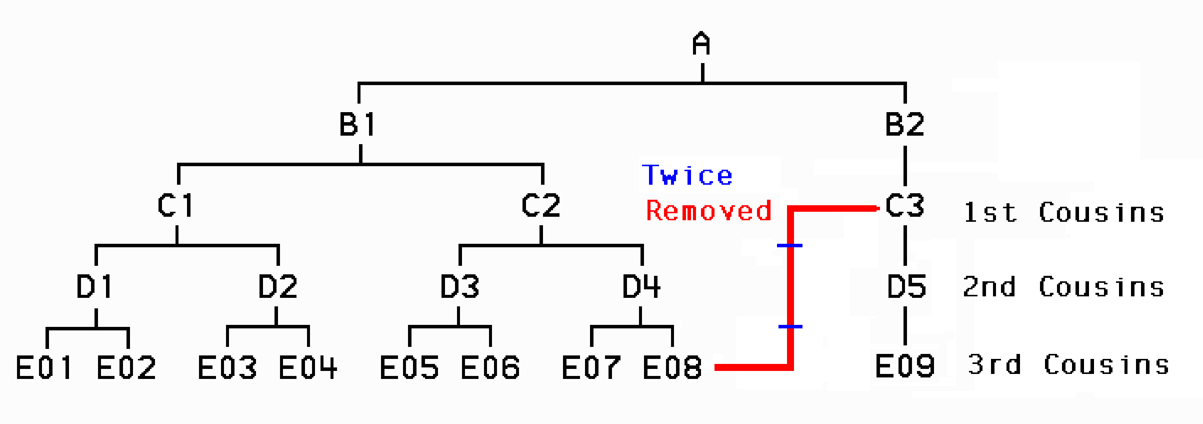
- $\forall_{x,y}$Aunt(x,y)⇒parent(p,y)∧Sibling(p,x)∧Gender(x,'Female')

X is aunt of Y when some ‘p’ is parent of Y and ‘p’ is sibling of X and X’s gender is female.

- $\forall_{x,y}$Uncle(x,y)⇒parent(p,y)∧Sibling(p,x)∧Gender(a,'Male')

X is uncle of Y when some ‘p’ is parent of Y and ‘p’ is sibling of X and X’s gender is male.

b) The definition of m^{th} cousin n times removed can be described as below.



We know that

First_Cousin=>children of siblings.

Second_Cousin=>children of first_cousins

M^{th} Cousin=>children of $(M-1)^{th}$ cousins

Removing can be expresses as, if two cousins are from different generations then they are removed once for each generation. For example, you can see it from the above picture that C3 and E08 are from different generations, and it describes that C3’s first cousins twice removed.

First-Order-Logic: To express m^{th} cousin, I’m considering the siblings as 0^{th} cousins.

$\forall_{x,y}$Cousin(a,b,m)⇒(greater(m,0)∧parent(p,x)∧parent(q,y)∧Cousin(p,q,m-1))∨(equal(m,0)∧Sibling(x,y))

$\forall_{x,y}$Cousin(x,y,m) ⇒ Cousin(y,x,m)

Now I will add a predicate n which represents “removed n times”.

$\forall_{x,y}$Cousin(x,y,m,n)⇒(greater(n,0)∧(parent(p,x)∧Cousin(p,y,m,n-1))∨(parent(p,y)∧Cousin(p,x,m,n-1)))∨(equal(n,0)∧Cousin(x,y,m))

$\forall_{x,y}$Cousin(x,y,m,n) ⇒ Cousin(y,x,m,n)

2. BAYESIAN NETWORKS:

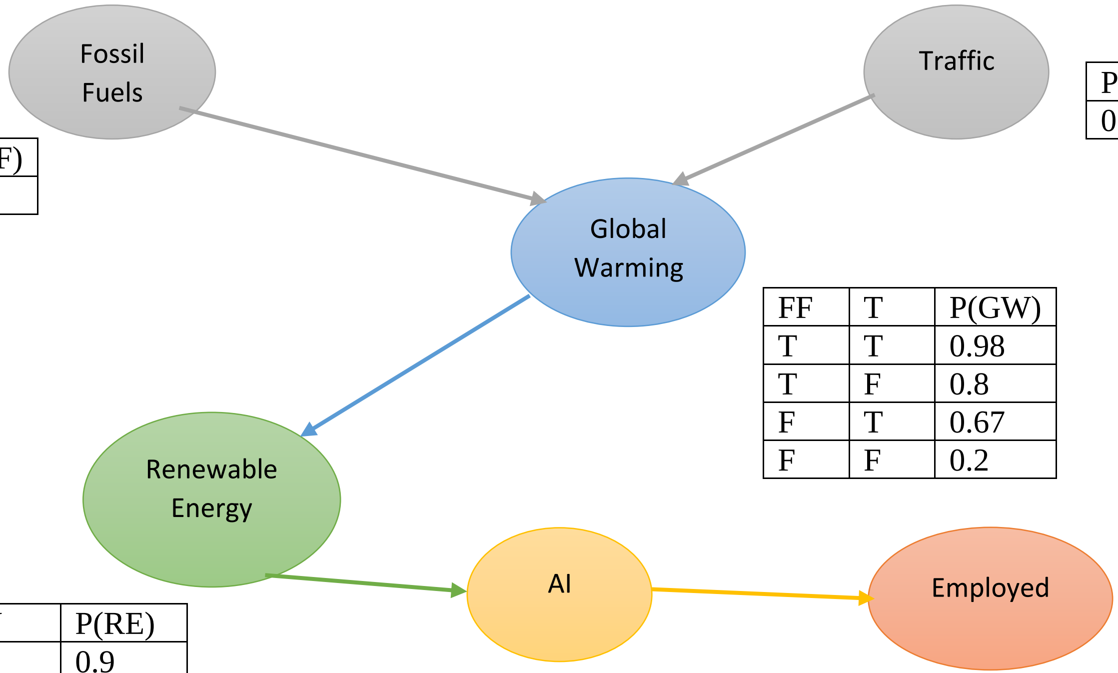
The given random variables are

- Fossil Fuels
- AI
- Global Warming
- Renewable Energy
- Traffic
- Employed

Let me construct a model of the world with these variables.

- Consider the Fossil Fuels and Traffic, these causes Global Warming.
- To control the Global Warming, we can use the Renewable Energy resources.
- This Renewable Energy sector can be improved with the help of AI technology.
- The AI can be developed by the Employed in the renewable energy sector.

The visual description and conditional probability tables are as below.



P(FF)
0.8

P(T)
0.6

FF	T	P(GW)
T	T	0.98
T	F	0.8
F	T	0.67
F	F	0.2

GW	P(RE)
T	0.9
F	0.4

RE	P(AI)
T	0.5
F	0.1

AI	P(Emp)
T	0.3
F	0.01

Setting the nodes with BayesNode:

```
# creating bayes nodes for the network.
GlobalWarming_node = BayesNode('GlobalWarming', ['FossilFuels', 'Traffic'],
                  {(True, True): 0.98,(True, False): 0.80, (False, True): 0.67, (False, False): 0.20})

RenewableEnergy_node = BayesNode('RenewableEnergy', ['GlobalWarming'], {True: 0.90, False: 0.40})

AI_node =BayesNode('AI', ['RenewableEnergy'], {True: 0.5, False: 0.1})

Emp_node =BayesNode('Employed', ['AI'], {True: 0.3, False: 0.01})


FossilFules_node = BayesNode('FossilFuels', '', 0.8)
Traffic_node = BayesNode('Traffic', '', 0.6)
```

Creating the Bayes Network with the help of BayesNet function:

```
T, F = True, False
# creating a Bayes Network with the help of predescribed nodes.
bayesNet = BayesNet([
            ('FossilFuels', '', 0.8),
            ('Traffic', '', 0.6),
            ('GlobalWarming', 'FossilFuels Traffic',
             {(T, T): 0.98, (T, F): 0.80, (F, T): 0.67, (F, F): 0.27}),
            ('RenewableEnergy', 'GlobalWarming', {T: 0.90, F: 0.40}),
            ('AI', 'RenewableEnergy', {T: 0.5, F: 0.1}),
            ('Employed', 'AI', {T: 0.5, F: 0.01})
            ])
```

Query: Evaluate the probability of Fossil Fuels in respect of Traffic -> False, AI -> True, Renewable Energy -> True.

```
print("Query :FossilFuels (Traffic -- False , AI -- True , Renewable Enery -- True)")
Query = enumeration_ask('FossilFuels',{'Traffic': False, 'AI': True, 'RenewableEnergy': True}, bayesNet)
print (Query.show_approx())
```

The output generated after querying

```
In [47]: runfile('C:/Users/reddy/AIMA/aima-python/
A2_COMP9016_Yarram_Jaipal_R00208981.py', wdir='C:/Users/reddy/AIMA/aima-
python')
Reloaded modules: utils, probabilistic_learning, learning, ipythonblocks,
agents, probability

 ******************** BAYESIAN NETWORKS *****************************

********* Renewable Energy Node *************
('RenewableEnergy', 'GlobalWarming')
0.4

********* Global Worming Node *************
('GlobalWarming', 'FossilFuels Traffic')
0.020000000000000018

******** Bayes Network ************
BayesNet([('FossilFuels', ''), ('Traffic', ''), ('GlobalWarming',
'FossilFuels Traffic'), ('RenewableEnergy', 'GlobalWarming'), ('AI',
'RenewableEnergy'), ('Employed', 'AI')])

******** Query ****************
Query :FossilFuels (Traffic -- False , AI -- True , Renewable Enery --
True)
False: 0.143, True: 0.857
```

Bayesian networks:

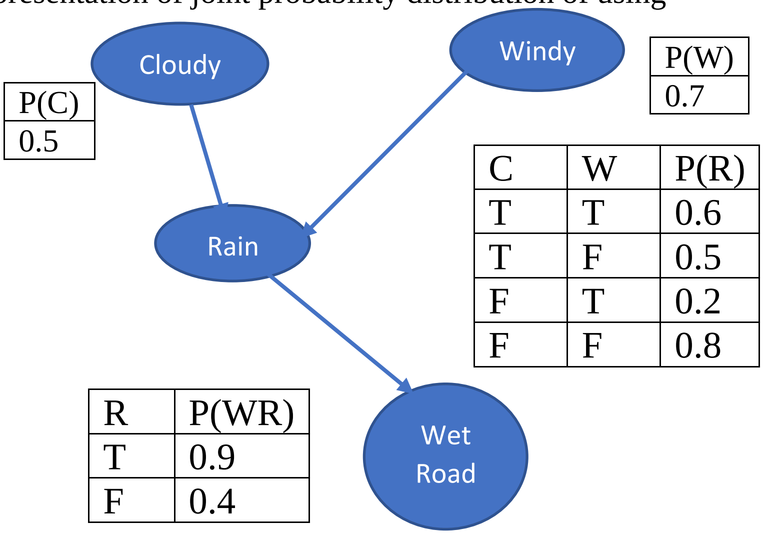
- It is a Probabilistic graphical model
- It applies Bayes Theorem for conditionally dependent and conditionally independent relationship between random nodes.
- It gives the compact, graphical representation of joint probability distribution of using conditional independence.

Pros:

- Structural Learning
- Fast response
- Decision analysis

Cons:

- Suitable for small dataset
- Not supported for loop networks



P(C)
0.5

P(W)
0.7

C	W	P(R)
T	T	0.6
T	F	0.5
F	T	0.2
F	F	0.8

R	P(WR)
T	0.9
F	0.4

Example:

The probability of cloudy 0.5 and probability of windy is 0.7.

If both are high, then the probability of road getting wet is high

3. LEARNING: DEVELOPING AND EVALUATING A MODEL

DATA:

Dataset: (https://archive.ics.uci.edu/ml/datasets/Lenses)

Dataset Information :

- The examples are complete and noise free.
- The examples highly simplified the problem.
- The attributes do not fully describe all the factors affecting the decision as to which type,
- if any, to fit.

Attribute Information:

2 Classes

- 1: The patient should be fitted with hard contact lenses,
- 2: The patient should be fitted with soft contact lenses,
- 3: The patient should not be fitted with contact lenses.

1. Age of the patient: (1) young, (2) pre-presbyopic, (3) presbyopic
2. Spectacle prescription: (1) myope, (2) hypermetrope
3. Astigmatic: (1) no, (2) yes
4. Tear production rate: (1) reduced, (2) normal

Implementing the Naïve Bayes Learner

1. Load the data Lenses and calculate the prior probabilities for each class

```
for target in target_distribution.dictionary.keys():
    print ('"Prior" probabilities for each of the classes {} is {}'.format(target, target_distribution[target]))
```

2. Calculate the probability of evidence

```
 # probability of evidence
print("\n\n2. Probability of evidence.")
for f in lenses.inputs:
    print("\nProbability of evidence for unique values of feature {} is: ".format(f))
    for val in FDistribution[f].dictionary.keys():
          print(" Value: {} ; Probabilty: {}".format(val,FDistribution[f][val]))
```

3. Calculate probability of likelihood of evidence.

```
   # probability of likelihood of evidences
print("\n\n3. Probability of likelihood of evidences (numerator).")
for featuredata in attributedistribution:
    print("\nFeature {} with class {} :".format(featuredata[1],featuredata[0]))
    for unique in attributedistribution[featuredata].dictionary.keys():
        print("Likelihood of evidence for value {}: {}".format(unique,attributedistribution[featuredata][unique]))
```

Output:

```
Probability of evidence for unique values of feature 1 is:
 Value: 1 ; Probabilty: 0.3333333333333333
 Value: 2 ; Probabilty: 0.3333333333333333
 Value: 3 ; Probabilty: 0.3333333333333333

Probability of evidence for unique values of feature 2 is:
 Value: 1 ; Probabilty: 0.5
 Value: 2 ; Probabilty: 0.5

Probability of evidence for unique values of feature 3 is:
 Value: 1 ; Probabilty: 0.5
 Value: 2 ; Probabilty: 0.5

Probability of evidence for unique values of feature 4 is:
 Value: 1 ; Probabilty: 0.5
 Value: 2 ; Probabilty: 0.5


3. Probability of likelihood of evidences (numerator).

Feature 0 with class 1 :
Likelihood of evidence for value 1: 0.03571428571428571
Likelihood of evidence for value 2: 0.03571428571428571
Likelihood of evidence for value 3: 0.03571428571428571
Likelihood of evidence for value 4: 0.07142857142857142
Likelihood of evidence for value 5: 0.03571428571428571
Likelihood of evidence for value 6: 0.03571428571428571
Likelihood of evidence for value 7: 0.03571428571428571
Likelihood of evidence for value 8: 0.07142857142857142
Likelihood of evidence for value 9: 0.03571428571428571
Likelihood of evidence for value 10: 0.03571428571428571
Likelihood of evidence for value 11: 0.03571428571428571
Likelihood of evidence for value 12: 0.07142857142857142
Likelihood of evidence for value 13: 0.03571428571428571
Likelihood of evidence for value 14: 0.03571428571428571
Likelihood of evidence for value 15: 0.03571428571428571
```

```
******************** NAIVE BAYES LEARNER ********************
Dataset - attrs: [0, 1, 2, 3, 4, 5]
Dataset - target: 5
Dataset - inputs: [0, 1, 2, 3, 4]
"Prior" probabilities for each of the classes 1 is 0.18518518518518517
"Prior" probabilities for each of the classes 2 is 0.2222222222222222
"Prior" probabilities for each of the classes 3 is 0.5925925925925926


2. Probability of evidence.

Probability of evidence for unique values of feature 0 is:
 Value: 1 ; Probabilty: 0.041666666666666664
 Value: 2 ; Probabilty: 0.041666666666666664
 Value: 3 ; Probabilty: 0.041666666666666664
 Value: 4 ; Probabilty: 0.041666666666666664
 Value: 5 ; Probabilty: 0.041666666666666664
 Value: 6 ; Probabilty: 0.041666666666666664
 Value: 7 ; Probabilty: 0.041666666666666664
 Value: 8 ; Probabilty: 0.041666666666666664
 Value: 9 ; Probabilty: 0.041666666666666664
 Value: 10 ; Probabilty: 0.041666666666666664
 Value: 11 ; Probabilty: 0.041666666666666664
 Value: 12 ; Probabilty: 0.041666666666666664
 Value: 13 ; Probabilty: 0.041666666666666664
 Value: 14 ; Probabilty: 0.041666666666666664
 Value: 15 ; Probabilty: 0.041666666666666664
 Value: 16 ; Probabilty: 0.041666666666666664
 Value: 17 ; Probabilty: 0.041666666666666664
 Value: 18 ; Probabilty: 0.041666666666666664
 Value: 19 ; Probabilty: 0.041666666666666664
 Value: 20 ; Probabilty: 0.041666666666666664
 Value: 21 ; Probabilty: 0.041666666666666664
 Value: 22 ; Probabilty: 0.041666666666666664
 Value: 23 ; Probabilty: 0.041666666666666664
 Value: 24 ; Probabilty: 0.041666666666666664
```

Naive Bayes Classifier:

- This is probabilistic classifier, the main objective is to process, analyze and categorize the data.
- Each feature makes equal contribution to the target class.
- Each feature is independent and equally contributes to the probability.
- Processes the training dataset to calculate the conditional probabilities and class probabilities.
- Easy to implement and fast computation and performs well on larger data.

There are 3 types of naïve Bayes classifier

- Multinomial Naïve Bayes
- Bernoulli Naïve Bayes
- Gaussian Naïve Bayes

Example:

Climate	Temperature	windy	Play Football
Sunny	Hot	TRUE	Yes
Rainy	Mild	TRUE	No
Rainy	Hot	FALSE	Yes
Sunny	Cool	TRUE	Yes

- We can find out whether the day is preferable to play football by classifying the data.
- The columns are the features, and the rows are the individual entries. We can use the
- Naïve Bayes classifier to predict the day which is preferable to play football using Bayes
- Theorem.

Probability:

Calculate a specific event occurred, given certain of attempts

Probability = events / outcomes

Conditional Independence:

We have a dataset with set of class (A), and we want to classify an item with a set of features (F)

Conditional probability given to the (F): P(A/F) = P(F/A) * P(A) / P(F)

Will have to do this process for every class:

The calculation of the conditional probability then depends on the calculation of the

following:

The probability of Class (A, B, C) in the dataset.

The conditional probability of each feature occurring in an item classified in Class.

The probabilities of each individual feature.

Bayes theorem:

Naïve Bayes is classification algorithm: to predict the class membership of unclassified data

P(A/B) = P(B/A) * P(A) / P(B)

P(A/B): Conditional probability: probability of A given B

P(B/A): Conditional probability: probability of B given A

P(A) and P(B): Probability of A and B (A and B are events

The conditional Probability is depending on:

- The probability of class in a dataset.
- The conditional probability of each feature from class.
- The probability of each feature.

References:

https://github.com/aimacode/aima-python

https://en.wikipedia.org/wiki/Ancestor

https://www.tedpack.org/cousins.html

https://www.sciencedirect.com/topics/engineering/naive-bayes-classifier

https://towardsdatascience.com/naive-bayes-classifier-81d512f50a7c