Name: Pratik Antoni Patekar

AI 1 Assignment 4

Task 1: Consider the given joint probability distribution for a domain of two variables (Color, Vehicle):

	Color = Red	Color = Green	Color = Blue
Vehicle = Car	**0.1299**	**0.0195**	**0.0322**
Vehicle = Van	**0.1681**	**0.0252**	**0.0417**
Vehicle = Truck	**0.1070**	**0.0160**	**0.0265**
Vehicle = SUV	**0.3103**	**0.0465**	**0.0769**

Part a: Calculate P (Color is not Green | Vehicle is Truck)

Part b: Prove that Vehicle and Color are totally independent from each other
Solution:

Part a: $P(\text{Color} = \neg \text{Green} \mid \text{Vehicle} = \text{Truck})$

$$= P(\text{Color} = \text{Red} \mid \text{Vehicle} = \text{Truck}) + P(\text{Color} = \text{Blue} \mid \text{Vehicle} = \text{Truck})$$

$$= \frac{P(\text{Color} = \text{Red} \wedge \text{Vehicle} = \text{Truck})}{P(\text{Vehicle} = \text{Truck})} + \frac{P(\text{Color} = \text{Blue} \wedge \text{Vehicle} = \text{Truck})}{P(\text{Vehicle} = \text{Truck})}$$

$$= \frac{0.1070}{0.1070 + 0.0160 + 0.0265} + \frac{0.0265}{0.1070 + 0.0160 + 0.0265}$$

$$= 0.893$$

Part b: Prove 'Vehicle' & 'Color' are independent of each other.

To do so, we need to prove,

$$P(\text{Vehicle} \mid \text{Color}) = P(\text{Vehicle})$$

OR

$$P(\text{Color} \mid \text{Vehicle}) = P(\text{Color})$$

Name: Pratik Antoni Patekar

For Color = ¬Green and Vehicle = Car :

$P(\text{Color} = \neg\text{Green} \mid \text{Vehicle} = \text{Truck}) = 0.893$ — (from part a)

$P(\text{Color} = \neg\text{Green}) = 1 - P(\text{Color} = \text{Green}) = 1 - (0.0195 + 0.0252 + 0.0160 + 0.0465)$

$= 0.8928 \simeq 0.893$

We can see that,

$P(\text{Color} = \neg\text{Green} \mid \text{Vehicle} = \text{Truck}) = P(\text{Color} = \text{Green})$

As both the values are same, we can say that the variables color and vehicle are independent of each other.

Task 2:
In a certain probability problem, we have 11 variables: A, B1, B2, ..., B10.

- **Variable A has 7 values.**
- **Each of variables B_1, ..., B_{10} have 8 possible values. Each B_i is conditionally independent of all other 9 B_j variables (with j != i) given A.**

Based on these facts:
Part a: How many numbers do you need to store in the joint distribution table of these 11 variables?
Part b: What is the most space-efficient way (in terms of how many numbers you need to store) representation for the joint probability distribution of these 11 variables? How many numbers do you need to store in your solution? Your answer should work with any variables satisfying the assumptions stated above.
Part c: Does this scenario follow the Naive-Bayes model?
Solution:

Part a: Given information, A – 7 values & $B_n \to$ 8 values each where n = 1 to 10

∴ # of values to be stored for JPD (theoritically) $= 7 \times 8^{10} = 7.51 \times 10^9$ (approx)

And pratically $= (7 \times 8^{10}) - 1$.

Part b: It is given that B_i is conditionally independent of B_j (where $j \neq i$) given A.

$$\therefore P(B_1, B_2, B_3, \ldots, B_{10} \mid A) = \prod_{i=1}^{10} P(B_i \mid A) \quad \text{— (1)}$$

Name: Pratik Antoni Patekar

$\therefore$ Using product rule we can say that,

$$P(A, B_1, B_2, B_3, \ldots, B_{10}) = P(B_1, B_2, B_3, \ldots, B_{10} \mid A) \cdot P(A)$$

$$= \left(\prod_{i=1}^{10} P(B_i \mid A)\right) \cdot P(A) \quad \text{—— (from ①)}$$

$$\left(\prod_{i=1}^{10} P(B_i \mid A)\right) \cdot P(A)$$

Each B_i variable can take 8 values & A can take 7 values.

$\therefore$ # of values to store for each $P(B_i \mid A) = 8 \times 7$

And as there are 10 B_i variables

$\therefore$ Total # of values to be stored for $P(B_i \mid A)$

$= 8 \times 7 \times 10$ —— ② (theory)

$= 7 \times 7 \times 10$ —— ④ (pratice)

A variable can take 7 values

$\therefore$ # of values to store for $P(A) = 7$ —— ③ (theory)

$= 6$ —— ⑤ (practice)

$\therefore$ Total # of values to be stored $= (8 \times 7 \times 10) + 7$ (from ② & ③)

$= 570$ (theoritically)

And practically $= (7 \times 7 \times 10) + 6 = 496$ (from ④ & ⑤)

Name: Pratik Antoni Patekar

Part c:

Here we know that the variables B_1, B_2,...., B_{10} are independent of each other given A. In a way, we can say that the effects (B_1, B_2,...., B_{10}) are independent of each other given the cause (A). Thus we can say that this scenario follows the Naive Bayes model.

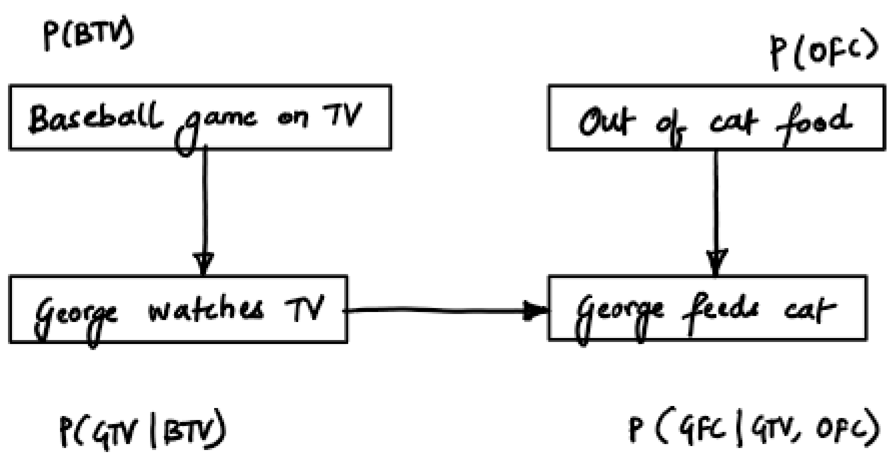
Task 3:

George doesn't watch much TV in the evening, unless there is a baseball game on. When there is baseball on TV, George is very likely to watch. George has a cat that he feeds most evenings, although he forgets every now and then. He's much more likely to forget when he's watching TV. He's also very unlikely to feed the cat if he has run out of cat food (although sometimes he gives the cat some of his own food). Design a Bayesian network for modeling the relations between these four events:

- **baseball_game_on_TV**
- **George_watches_TV**
- **out_of_cat_food**
- **George_feeds_cat**

Your task is to connect these nodes with arrows pointing from causes to effects. No programming is needed for this part, just include an electronic document (PDF, Word file, or OpenOffice document) showing your Bayesian network design.

Solution:



Name: Pratik Antoni Patekar

Task 4:

For the Bayesian network of previous task, the text file at this link contains training data from every evening of an entire year. Every line in this text file corresponds to an evening, and contains four numbers. Each number is a 0 or a 1. In more detail:

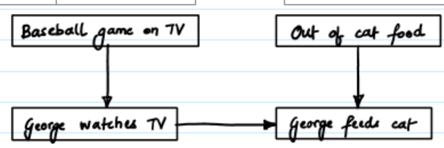
- The first number is 0 if there is no baseball game on TV, and 1 if there is a baseball game on TV.
- The second number is 0 if George does not watch TV, and 1 if George watches TV.
- The third number is 0 if George is not out of cat food, and 1 if George is out of cat food.
- The fourth number is 0 if George does not feed the cat, and 1 if George feeds the cat.

Based on the data in this file, determine the probability table for each node in the Bayesian network you have designed for Task 3. You need to include these four tables in the drawing that you produce for question 3. You also need to submit the code/script that computes these probabilities.

Solution:

P(baseball_game_on_TV = True)	0.3041095890410959

P(out_of_cat_food = True)	0.16986301369863013



P(George_watches_TV)	**baseball_game_on_TV = True**	**baseball_game_on_TV = False**
True	0.9279279279279279	0.11811023622047244
False	0.07207207207207211	0.8818897637795275

P(George_watches_TV = True | baseball_game_on_TV = True) = 0.9279279279279279
P(George_watches_TV = False | baseball_game_on_TV = True) = 0.07207207207207211
P(George_watches_TV = True | baseball_game_on_TV = False) = 0.11811023622047244
P(George_watches_TV = False | baseball_game_on_TV = False) = 0.8818897637795275

	out_of_cat_food = True	**out_of_cat_food = True**	**out_of_cat_food = False**	**out_of_cat_food = False**
P(George_feeds_cat)	**George_watches_TV = True**	**George_watches_TV = False**	**George_watches_TV = True**	**George_watches_TV = False**
True	0.041666666666666664	0.3157894736842105	0.7064220183486238	0.9587628865979382
False	0.9583333333333334	0.6842105263157895	0.29357798165137616	0.04123711340206182

P(George_feeds_cat = True | (George_watches_TV = True , out_of_cat_food = True))) = 0.041666666666666664
P(George_feeds_cat = False | (George_watches_TV = True , out_of_cat_food = True))) = 0.9583333333333334
P(George_feeds_cat = True | (George_watches_TV = False , out_of_cat_food = True))) = 0.3157894736842105
P(George_feeds_cat = False | (George_watches_TV = False , out_of_cat_food = True))) = 0.6842105263157895
P(George_feeds_cat = True | (George_watches_TV = True , out_of_cat_food = False))) = 0.7064220183486238
P(George_feeds_cat = False | (George_watches_TV = True , out_of_cat_food = False))) = 0.29357798165137616
P(George_feeds_cat = True | (George_watches_TV = False , out_of_cat_food = False))) = 0.9587628865979382
P(George_feeds_cat = False | (George_watches_TV = False , out_of_cat_food = False))) = 0.04123711340206182

Following is the Python 3 script that calculated the above values:

1. Code to read the file and store as pandas dataframe:

```
import pandas as pd

# Read file contents and store them in pandas dataframe with required headers
headers = ['baseball_game_on_TV', 'George_watches_TV', 'out_of_cat_food',
'George_feeds_cat']
data = []

with open('training_data.txt', 'r') as f:
   line = f.readline().rstrip('\n')
   data.append(line.split('     ')[1:])
   while line:
      line = f.readline().rstrip('\n')
      if line != "":
         data.append(line.split('     ')[1:])
```

```
df = pd.DataFrame(data, columns = headers)
df
```

	baseball_game_on_TV	George_watches_TV	out_of_cat_food	George_feeds_cat
0	0	0	1	1
1	0	0	0	1
2	0	0	0	1
3	0	0	0	1
4	0	0	0	1
...	...	...	...	...
360	0	0	1	0
361	0	0	0	1
362	1	0	1	0
363	0	0	0	1
364	0	0	0	1

365 rows × 4 columns

(Above is the screenshot of the contents of df dataframe)

2. Functions to calculate the probabilities:

```
# Function to calculate the prior probabilities
def calc_prior_prob(df, var_name):
    count_true = 0
    tot_count = 0
    p1 = len(df[df[var_name] == '1'])/ len(df[var_name])

    return p1

# Function to calculate the conditional probabilities
def calc_cond_prob(df, var_name, given_var_name1, given_var_name2 = 'any'):
    if given_var_name2 == 'any':
        p1 = (len(df[(df[given_var_name1] == '1') & (df[var_name] ==
'1')]))/(len(df[df[given_var_name1] == '1']))
        p2 = (len(df[(df[given_var_name1] == '0') & (df[var_name] ==
'1')]))/(len(df[df[given_var_name1] == '0']))
        return p1, p2
    else:
        p1 = (len(df[(df[given_var_name1] == '1') & (df[given_var_name2] == '1') & (df[var_name]
== '1')]))/(len(df[(df[given_var_name1] == '1') & (df[given_var_name2] == '1')]))
```

```
    p2 = (len(df[(df[given_var_name1] == '0') & (df[given_var_name2] == '1') & (df[var_name]
== '1')]))/(len(df[(df[given_var_name1] == '0') & (df[given_var_name2] == '1')]))
    p3 = (len(df[(df[given_var_name1] == '1') & (df[given_var_name2] == '0') & (df[var_name]
== '1')]))/(len(df[(df[given_var_name1] == '1') & (df[given_var_name2] == '0')]))
    p4 = (len(df[(df[given_var_name1] == '0') & (df[given_var_name2] == '0') & (df[var_name]
== '1')]))/(len(df[(df[given_var_name1] == '0') & (df[given_var_name2] == '0')]))
    return p1, p2, p3, p4
```

3. Code to calculate the prior probabilities and conditional probabilities using the above functions and dataframe read from the text file:

```
p1 = calc_prior_prob(df, 'baseball_game_on_TV')
print('P(baseball_game_on_TV) =', p1)
p2 =  calc_prior_prob(df, 'out_of_cat_food')
print('P(out_of_cat_food) =', p2)

p3, p4 = calc_cond_prob(df, 'George_watches_TV', 'baseball_game_on_TV')
print('P(George_watches_TV = True | baseball_game_on_TV = True) =', p3,
'\nP(George_watches_TV = False | baseball_game_on_TV = True) =', 1-p3)
print('P(George_watches_TV = True | baseball_game_on_TV = False) =', p4,
'\nP(George_watches_TV = False | baseball_game_on_TV = False) =', 1-p4)
print('\n')

p5, p6, p7, p8 = calc_cond_prob(df, 'George_feeds_cat', 'George_watches_TV',
'out_of_cat_food')
print('P(George_feeds_cat = True | (George_watches_TV = True , out_of_cat_food = True))) = ',
p5)
print('P(George_feeds_cat = False | (George_watches_TV = True , out_of_cat_food = True))) =
', 1-p5)
print('P(George_feeds_cat = True | (George_watches_TV = False , out_of_cat_food = True))) =
', p6)
print('P(George_feeds_cat = False | (George_watches_TV = False , out_of_cat_food = True))) =
', 1-p6)
print('P(George_feeds_cat = True | (George_watches_TV = True , out_of_cat_food = False))) =
', p7)
print('P(George_feeds_cat = False | (George_watches_TV = True , out_of_cat_food = False))) =
', 1-p7)
print('P(George_feeds_cat = True | (George_watches_TV = False , out_of_cat_food = False))) =
', p8)
print('P(George_feeds_cat = False | (George_watches_TV = False , out_of_cat_food = False)))
= ', 1-p8)
```

Name: Pratik Antoni Patekar

Final output:

```
P(baseball_game_on_TV) = 0.3041095890410959
P(out_of_cat_food) = 0.16986301369863013
P(George_watches_TV = True | baseball_game_on_TV = True) = 0.9279279279279279
P(George_watches_TV = False | baseball_game_on_TV = True) = 0.07207207207207211
P(George_watches_TV = True | baseball_game_on_TV = False) = 0.11811023622047244
P(George_watches_TV = False | baseball_game_on_TV = False) = 0.8818897637795275
P(George_feeds_cat = True | (George_watches_TV = True , out_of_cat_food = True))) =  0.041666666666666664
P(George_feeds_cat = False | (George_watches_TV = True , out_of_cat_food = True))) =  0.9583333333333334
P(George_feeds_cat = True | (George_watches_TV = False , out_of_cat_food = True))) =  0.3157894736842105
P(George_feeds_cat = False | (George_watches_TV = False , out_of_cat_food = True))) =  0.6842105263157895
P(George_feeds_cat = True | (George_watches_TV = True , out_of_cat_food = False))) =  0.7064220183486238
P(George_feeds_cat = False | (George_watches_TV = True , out_of_cat_food = False))) =  0.29357798165137616
P(George_feeds_cat = True | (George_watches_TV = False , out_of_cat_food = False))) =  0.9587628865979382
P(George_feeds_cat = False | (George_watches_TV = False , out_of_cat_food = False))) =  0.04123711340206182
```

Task 5: Given the network obtained in the previous two tasks, calculate P (Baseball Game on TV | not(George Feeds Cat)) using Inference by Enumeration.
Solution:

To find the probability from Bayesian network we use the following formula:

$$\boxed{P(X_1, \ldots, X_i) = \prod_{i=1}^{n} P(X_i \mid parents(X_i))}$$

(Image reference: Professor Vamsikrishna Gopikrishna lecture slides)

Note: The values from the conditional probability tables from the Bayesian network in the part 4, have been rounded up to 2 decimal places for the calculations below.

$$P(\text{Baseball\_game\_on\_TV} = \text{True} \mid \text{George\_feeds\_cat} = \text{False})$$

$$= \frac{P((\text{Baseball\_game\_on\_TV} = \text{True}), (\text{George\_feeds\_cat} = \text{False}))}{P(\text{George\_feeds\_cat} = \text{False})} = \frac{N}{D}.$$

Name: Pratik Antoni Patekar

$N = P((\text{Baseball\_game\_on\_TV}=\text{True}), (\text{George\_feeds\_cat} = \text{False}))$

$= P(BTV=True, GFC = False, GTV = True, OCF = True) +$
$P(BTV=True, GFC = False, GTV = True, OCF = False) +$
$P(BTV=True, GFC = False, GTV = False, OCF = True) +$
$P(BTV=True, GFC = False, GTV = False, OCF = False)$

$= P(BTV=True) \times P(GFC = False \mid GTV=True, OCF=True) \times P(GTV=True \mid BTV=True) \times P(OCF=True) +$
$P(BTV=True) \times P(GFC = False \mid GTV=True, OCF=False) \times P(GTV=True \mid BTV=True) \times P(OCF=False) +$
$P(BTV=True) \times P(GFC = False \mid GTV=False, OCF=True) \times P(GTV=False \mid BTV=True) \times P(OCF=True) +$
$P(BTV=True) \times P(GFC = False \mid GTV=False, OCF=False) \times P(GTV=False \mid BTV=True) \times P(OCF=False)$

$= 0.30 \times 0.96 \times 0.93 \times 0.17 +$
$0.30 \times 0.29 \times 0.93 \times 0.83 +$
$0.30 \times 0.68 \times 0.07 \times 0.17 +$
$0.30 \times 0.04 \times 0.07 \times 0.83$

$= 0.1158$

$D = P(GFC = False)$

$= P(GFC=False, GTV=True, BTV=True, OCF=True) +$
$P(GFC=False, GTV=True, BTV=True, OCF=False) +$
$P(GFC=False, GTV=True, BTV=False, OCF=True) +$
$P(GFC=False, GTV=True, BTV=False, OCF=False) +$
$P(GFC=False, GTV=False, BTV=True, OCF=True) +$
$P(GFC=False, GTV=False, BTV=True, OCF=False) +$
$P(GFC=False, GTV=False, BTV=False, OCF=True) +$
$P(GFC=False, GTV=False, BTV=False, OCF=False)$

$= P(GFC=False \mid GTV=True, OCF=True) \times P(GTV=True \mid BTV=True) \times P(BTV=True) \times P(OCF=True) +$
$P(GFC=False \mid GTV=True, OCF=False) \times P(GTV=True \mid BTV=True) \times P(BTV=True) \times P(OCF=False) +$
$P(GFC=False \mid GTV=True, OCF=True) \times P(GTV=True \mid BTV=False) \times P(BTV=False) \times P(OCF=True) +$
$P(GFC=False \mid GTV=True, OCF=False) \times P(GTV=True \mid BTV=False) \times P(BTV=False) \times P(OCF=False) +$
$P(GFC=False \mid GTV=False, OCF=True) \times P(GTV=False \mid BTV=True) \times P(BTV=True) \times P(OCF=True) +$
$P(GFC=False \mid GTV=False, OCF=False) \times P(GTV=False \mid BTV=True) \times P(BTV=True) \times P(OCF=False) +$
$P(GFC=False \mid GTV=False, OCF=True) \times P(GTV=False \mid BTV=False) \times P(BTV=False) \times P(OCF=True) +$
$P(GFC=False \mid GTV=False, OCF=False) \times P(GTV=False \mid BTV=False) \times P(BTV=False) \times P(OCF=False) +$

$= (0.96 \times 0.93 \times 0.30 \times 0.17) + (0.29 \times 0.93 \times 0.30 \times 0.83) +$
$(0.96 \times 0.12 \times 0.70 \times 0.17) + (0.29 \times 0.12 \times 0.70 \times 0.83) +$
$(0.68 \times 0.07 \times 0.30 \times 0.17) + (0.04 \times 0.07 \times 0.30 \times 0.83) +$
$(0.68 \times 0.88 \times 0.70 \times 0.17) + (0.04 \times 0.88 \times 0.70 \times 0.83)$

$= 0.2414 \approx 0.24$

$\therefore$ Req prob $= \frac{N}{D} = \frac{0.1158}{0.24} = 0.4796$

≈ 0.48

Task 6:

Class	A	B	C
X	1	2	1
X	2	1	2
X	3	2	2
X	1	3	3
X	1	2	1
Y	2	1	2
Y	3	1	1
Y	2	2	2
Y	3	3	1
Y	2	1	1

We want to build a decision tree that determines whether a certain pattern is of type X or type Y. The decision tree can only use tests that are based on attributes A, B, and C. Each attribute has 3 possible values: 1, 2, 3 (we do not apply any thresholding). We have the 10 training examples, shown on the table (each row corresponds to a training example). What is the information gain of each attribute at the root? Which attribute achieves the highest information gain at the root?

Name: Pratik Antoni Patekar

Attribute A :

$A \; [X, Y] = [5, 5] \longrightarrow H = -(5/10)\log_2(5/10) - (5/10)\log_2(5/10) = 1$

$A \xrightarrow{1} [3, 0]$, $A \xrightarrow{2} [1, 3]$, $A \xrightarrow{3} [1, 2]$

$H_1 = -(3/3)\log_2(3/3) - (0/3)\log_2(0/3) = 0$

$H_2 = -(1/4)\log_2(1/4) - (3/4)\log_2(3/4) = -(1/4)(-2) - (3/4)(-0.415) = 0.81125$

$H_3 = -(1/3)\log_2(1/3) - (2/3)\log_2(2/3) = (-1/3 \times -1.58) + (-2/3 \times -0.58) = 0.918$

Information gain for attribute A $= H - \sum_{i=1}^{L} (k_i/k)\, H_i$

$$= 1 - \left[\left(\frac{3}{10}\right)(0) + \left(\frac{4}{10}\right)(0.8112) + \left(\frac{3}{10}\right)(0.918)\right]$$

$$= 0.40012 \approx 0.4$$

Attribute B :

$B \; [5, 5] \longrightarrow H = -(5/10)\log_2(5/10) - (5/10)\log_2(5/10) = 1$

$B \xrightarrow{1} [1, 3]$, $B \xrightarrow{2} [3, 1]$, $B \xrightarrow{3} [1, 1]$

$H_1 = -(1/4)\log_2(1/4) - (3/4)\log_2(3/4) = 0.8112$

$H_2 = -(3/4)\log_2(3/4) - (1/4)\log_2(1/4) = 0.8112$

$H_3 = -(1/2)\log_2(1/2) - (1/2)\log_2(1/2) = 1$

$\therefore$ Information gain for attribute B

$$= H - \sum_{i=1}^{L} (k_i/k)\, H_i$$

$$= 1 - \left[\left(\frac{4}{10}\right)(0.8112) + \left(\frac{4}{10}\right)(0.8112) + \left(\frac{2}{10}\right)(1)\right]$$

$$= 0.15104$$

Name: Pratik Antoni Patekar

<u>Attribute c:</u>

c $[5, 5] \rightarrow \boxed{H = 1}$

$[2, 3]$

$H_1 = -(2/5)\log_2(2/5) - (3/5)\log_2(3/5)$

$= 0.97$

$[2, 2]$

$H_2 = -(2/4)\log_2(2/4) - (2/4)\log_2(2/4)$

$= 1$

$[1, 0]$

$H_3 = -(1/1)\log_2(1/1) - (0/1)\log_2(0/1)$

$= 0$

$\therefore$ Information gain for attribute c

$$= 1 - \left[\left(\frac{5}{10}\right)(0.97) + \left(\frac{4}{10}\right)(1) + \left(\frac{1}{10}\right)(0)\right]$$

$$= 0.115$$

Attribute	Information gain
A	0.4
B	0.15
C	0.11

$\because$ Attribute A has maximum information gain we can select it at root.