CSE 674 Project 1: Determining Probabilities of Handwriting Formations using PGMs

Mrudula Y
Department of Computer Science
University at Buffalo, SUNY
Buffalo, NY 14214
mrudulay@buffalo.edu

Abstract

In this project four tasks have been accomplished mainly to learn how to work with Bayesian Networks, pgmpy library, Inferences from Models and conversion of Bayesian Model to Markov Model. It has been observed that no matter how the dataset varies, if proper ancestral sampling is done and data is chosen using a discrete distribution sampler, we always get the same maximum probability setting of features.

1 Task 1

Evaluate pairwise correlations and independences that exist in the data. Note that we can determine whether xi and xj are independent by testing if p(xi, xj) = p(xi)p(xj), where the joint probability between a pair of variables can be determined from the tables as p(xi, xj) = p(xi|xj)p(xj)

1.1 Steps Followed to Accomplish Task 1

- Cross entropy metric was chosen to find how well the features are correlated with each other.
- Here correlation is used to know the causality between the features and in turn it also gives information about which features are independent of each other.
- Cross entropy is inversely related to correlation. [2]
- The following formula was used to calculate cross entropy -

$$H(p,q) = -\sum_x p(x)\ \log q(x).$$

[1]

- The p(x_i, x_j) in our project corresponds to p(x) and p(x_i)p(x_j) corresponds to q(x).
- As we have the marginal and conditional probabilities available, we can calculate the joint probability p(x_i, x_j) using the following formula -

$$P(x,y) = P(x \mid y)\, P(y)$$

- Cross entropy is calculated between p(x_i, x_j) and p(x_i)p(x_j) because of the fact the independence between two features can be known if joint probability is equal to the product of their individual marginal probabilities.
- A threshold of cross entropy was set to 2.0, to decide which features are least correlated. The pairs that had cross entropy values below 2.0 have very poor causal relationship

between each other that is are very loosely correlated.

1.2 Results of Task 1

Below is a table given that shows pairs that have high causal relationship and correlation between them. While calculating the cross entropy, conditional probabilities are used. In the table, feature and evidence feature mean that the conditional probability used while calculating cross entropy was the probability of the “feature” given the “evidence feature”.

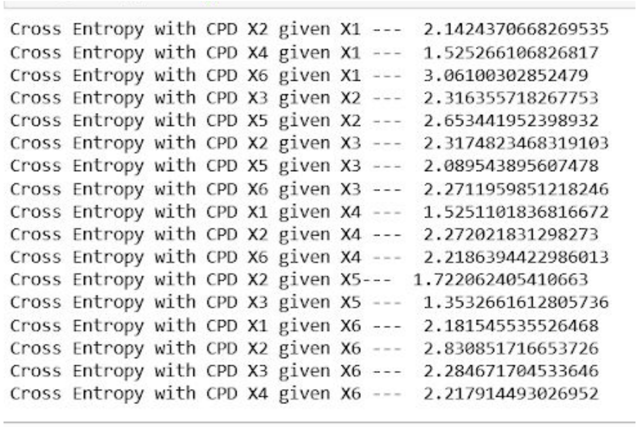
Table 1: Cross Entropy values in decreasing order of some highly correlated features

Feature	Evidence Feature	Cross Entropy Value
x3	x5	1.353
x4	x1	1.525
x2	x5	1.722

Table 2: Cross Entropy values in decreasing order of some loosely correlated features

Feature	Evidence Feature	Cross Entropy Value
x1	x6	3.061
x2	x6	2.830
x5	x2	2.653
x2	x3	2.317

Below Figure 1 shows the cross entropy values calculated between various pairs of features. Here we can observe that the cross entropy values for x3 given x5 is 1.353 but the cross entropy value of x5 given x3 is 2.089. There is a significant difference in the cross entropy value which shows that there is a higher chance that x5 causes x3 rather than the vice-versa. This is the reason why all possible combinations were calculated, to observe whether the cross entropy values vary significantly in any case or not.



```
Cross Entropy with CPD X2 given X1 ---  2.1424370668269535
Cross Entropy with CPD X4 given X1 ---  1.525266106826817
Cross Entropy with CPD X6 given X1 ---  3.06100302852479
Cross Entropy with CPD X3 given X2 ---  2.316355718267753
Cross Entropy with CPD X5 given X2 ---  2.653441952398932
Cross Entropy with CPD X2 given X3 ---  2.3174823468319103
Cross Entropy with CPD X5 given X3 ---  2.089543895607478
Cross Entropy with CPD X6 given X3 ---  2.2711959851218246
Cross Entropy with CPD X1 given X4 ---  1.5251101836816672
Cross Entropy with CPD X2 given X4 ---  2.272021831298273
Cross Entropy with CPD X6 given X4 ---  2.2186394422986013
Cross Entropy with CPD X2 given X5---  1.722062405410663
Cross Entropy with CPD X3 given X5 ---  1.3532661612805736
Cross Entropy with CPD X1 given X6 ---  2.181545535526468
Cross Entropy with CPD X2 given X6 ---  2.830851716653726
Cross Entropy with CPD X3 given X6 ---  2.284671704533646
Cross Entropy with CPD X4 given X6 ---  2.217914493026952
```

Figure 1: Cross entropy values between different pairs of features.
It shows the causal relationship between various features

2 Task 2

Construct a Bayesian network with the fewest number of edges that maximizes the likelihood. One approach is to use the results of the first task and start drawing links between the most correlated pairs of variables. We can construct several Bayesian networks and obtain a score for each of them. One way of scoring is to determine the likelihood the network assigns to samples generated (using ancestral sampling). Note that the dataset changes for each model. Based on your best model, describe what a high probability “th” looks like (in words as well as in image form). Describe some low probability “th” as well**.**

2.1 Steps Followed to Accomplish Task 2

- Marginal probabilities and the CPDs were put into lists so that the values can be used later while sampling data and creating the models.
- Five bayesian networks were manually constructed based on the cross entropy values obtained in task 1.
- For each of the five bayesian networks, 1000 data points were sampled using ancestral sampling (note: CPDs were given). The 1000 data points were sampled using a discrete distribution sampling method that randomly samples data from an array based on the probability distribution of those discrete data points in the array. The function used was from the numpy python library called random.choice().
- After sampling the data, the BayesianModel() function from the pgmpy library was used to construct a Bayesian model.
- The generated dataset which was a dataframe was passed to the K2 Score function available in the pgmpy library to calculate the K2 score for the model.
- This way five different datasets were created each of 1000 points and five K2 scores were obtained for each of the models.

2.2 Bayesian Networks Construction

The following figures show the manually generated bayesian Networks.

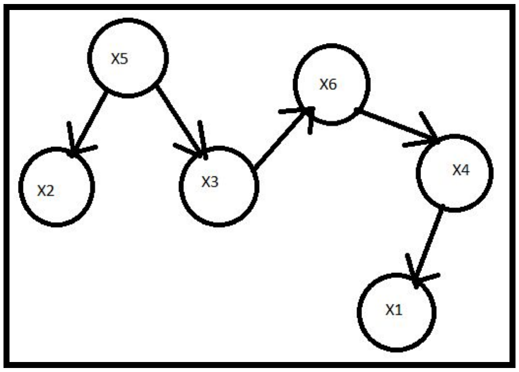


Figure 2: Bayesian Network 1

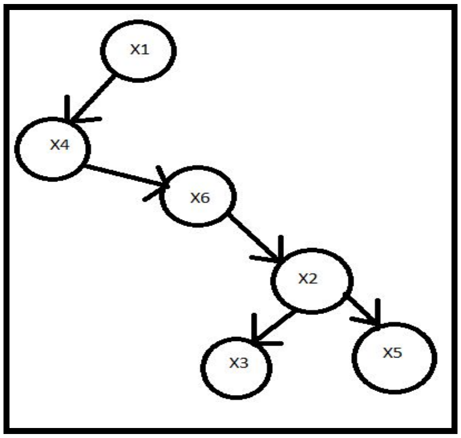


Figure 3: Bayesian Network 2

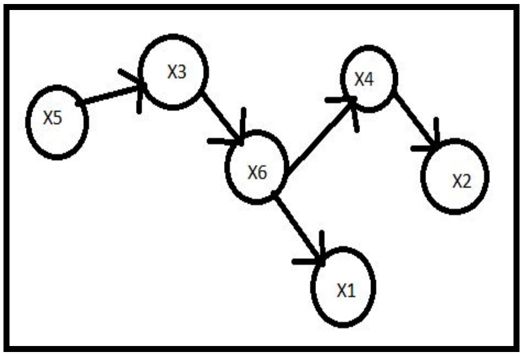


Figure 4: Bayesian Network 3

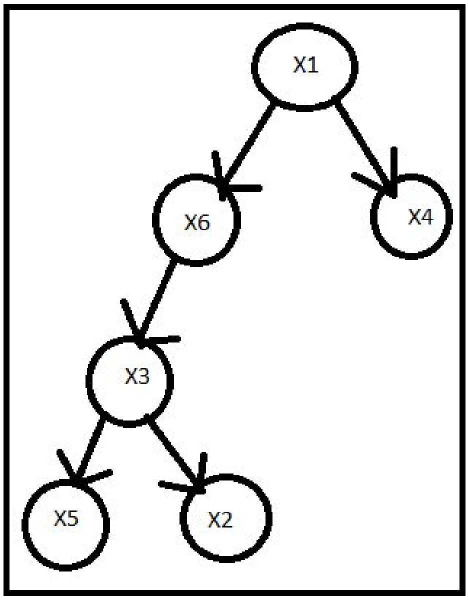


Figure 5: Bayesian Network 4

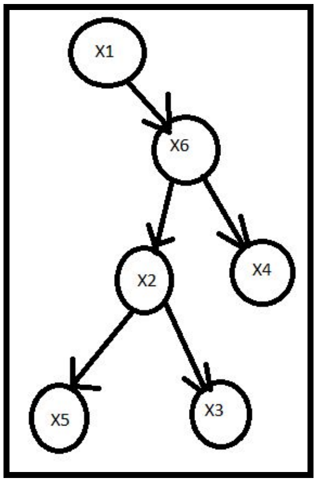


Figure 6: Bayesian Network 5

The joint probability equation for Figure 2 BN is -

$$p(x1, x2, x3, x4, x5) \ = \ p(x5)p(x1|x4)p(x4|x6)p(x6|x3)p(x3|x5)p(x2|x5)$$

The joint probability equation for Figure 3 BN is -

$$p(x1, x2, x3, x4, x5) \ = p(x1)p(x4|x1)p(x6|x4)p(x2|x6)p(x3|x2)p(x5|x2)$$

The joint probability equation for Figure 4 BN is -

$$p(x1, x2, x3, x4, x5) \ = p(x5)p(x1|x6)p(x3|x5)p(x6|x3)p(x4|x6)p(x2|x4)$$

The joint probability equation for Figure 5 BN is -

$$p(x1, x2, x3, x4, x5) \ = p(x1)p(x4|x3)p(x2|x3)p(x3|x6)p(x6|x1)p(x5|x1)$$

The joint probability equation for Figure 6 BN is -

$$p(x1, x2, x3, x4, x5) \ = p(x1)p(x6|x1)p(x2|x6)p(x5|x2)p(x3|x2)p(x4|x6)$$

2.3 Results of Task 2

2.3.1 Deciding the best bayesian model

Table 3: K2 Score of the constructed Bayesian Models. Highlighted: Best Bayesian Model

	BN 1	BN 2	BN 3	BN 4	BN5
K2 Score	-6480.311379	-6458.119469	-6209.38027	-6323.75429	-6424.04477

We can see from the table that the Bayesian Model 3 has the highest K2 Score. This means that this model best represents the relation between the various features and the causality between them.

2.3.2 High probability and low probability “th” based on best bayesian model

After getting the best bayesian model, the sampled dataset of this model is observed and the joint probability for all the 1000 possible combinations of x1-x6 values is calculated. The set of values x1-x6 take for which the obtained joint probability is the highest gives the most probable setting of these features. As these features represent different strokes made for writing “th”, we know that which kind of “th” mostly found. Some settings that had low joint probability values were also observed. These gave the “th” that with low probability.

Table 4: The most probable setting of features for “th” gives the high probability “th”

x1	x2	x3	x4	x5	x6
0	1	1	0	3	3

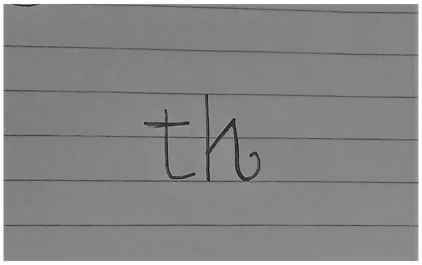
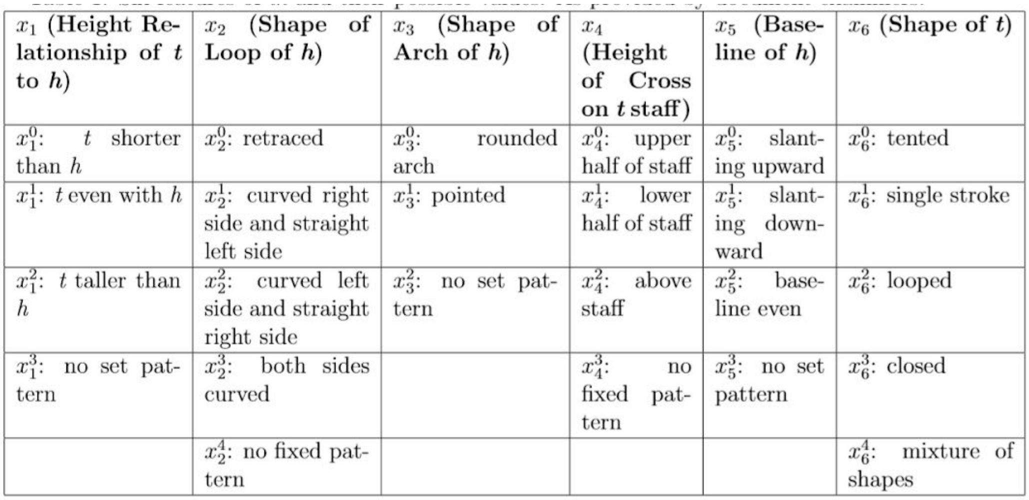


Figure 7: Image of most probable “th” with straight left, curved right and pointed arch h with t’s height shorter than h.

Table 5: The less probable settings of features for “th” gives the low probability “th”

x1	x2	x3	x4	x5	x6
3	1	1	0	3	0
3	4	1	0	3	0
1	1	0	0	1	4

The below figures tells the meaning of features from x1-x6 and their different values -

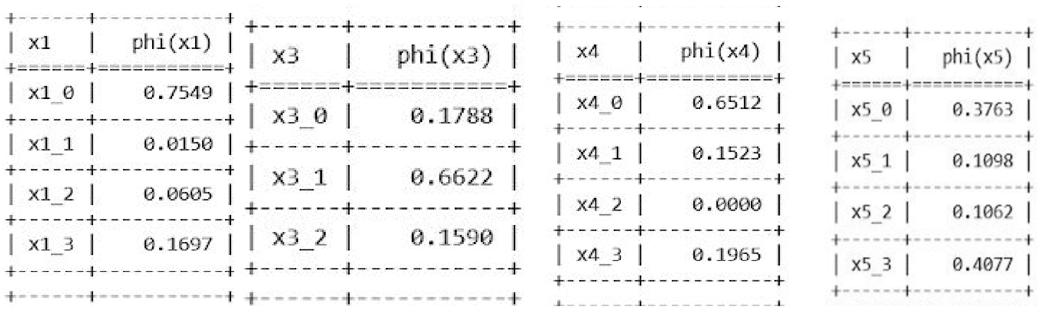


x_1 **(Height Relationship of *t* to *h*)**	x_2 **(Shape of Loop of *h*)**	x_3 **(Shape of Arch of *h*)**	x_4 **(Height of Cross on *t* staff)**	x_5 **(Baseline of *h*)**	x_6 **(Shape of *t*)**
x_1^0: t shorter than h	x_2^0: retraced	x_3^0: rounded arch	x_4^0: upper half of staff	x_5^0: slanting upward	x_6^0: tented
x_1^1: t even with h	x_2^1: curved right side and straight left side	x_3^1: pointed	x_4^1: lower half of staff	x_5^1: slanting downward	x_6^1: single stroke
x_1^2: t taller than h	x_2^2: curved left side and straight right side	x_3^2: no set pattern	x_4^2: above staff	x_5^2: baseline even	x_6^2: looped
x_1^3: no set pattern	x_2^3: both sides curved		x_4^3: no fixed pattern	x_5^3: no set pattern	x_6^3: closed
	x_2^4: no fixed pattern				x_6^4: mixture of shapes

Figure 8: Meaning of features represented by x1-x6

2.3.3 Inference from the best bayesian model

Variable Elimination function was used to get an inference object for the best model. The query function is then run on this inference object to infer the unknown conditional probabilities.



```
+------+-----------+
| x1   |   phi(x1) |
+======+===========+
| x1_0 |    0.7549 |
+------+-----------+
| x1_1 |    0.0150 |
+------+-----------+
| x1_2 |    0.0605 |
+------+-----------+
| x1_3 |    0.1697 |
+------+-----------+
```

```
+------+-----------+
| x3   |   phi(x3) |
+======+===========+
| x3_0 |    0.1788 |
+------+-----------+
| x3_1 |    0.6622 |
+------+-----------+
| x3_2 |    0.1590 |
+------+-----------+
```

```
+------+-----------+
| x4   |   phi(x4) |
+======+===========+
| x4_0 |    0.6512 |
+------+-----------+
| x4_1 |    0.1523 |
+------+-----------+
| x4_2 |    0.0000 |
+------+-----------+
| x4_3 |    0.1965 |
+------+-----------+
```

```
+------+-----------+
| x5   |   phi(x5) |
+======+===========+
| x5_0 |    0.3763 |
+------+-----------+
| x5_1 |    0.1098 |
+------+-----------+
| x5_2 |    0.1062 |
+------+-----------+
| x5_3 |    0.4077 |
+------+-----------+
```

(a) (b) (c) (d)

Figure 9: Few Inferences from the best Bayesian Model (a) marginal prob(x1) (b) p(x3|x2=0) (c) p(x4|x3=0,x2=4) (d) p(x5|x1=1)

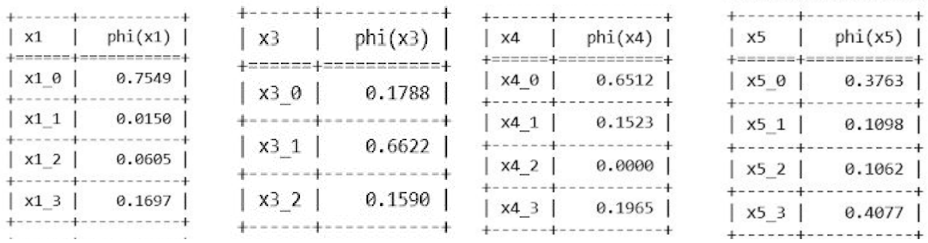
3 Task 3

Convert your best Bayesian network into a Markov network using moralization. Compare inferences using Bayesian network and the Markov network, in terms of computation time and accuracy.

3.1 Steps Followed to Accomplish Task 3

- The to_markov() method in Bayesian Model pgmpy library was used to convert the best Bayesian model obtained in task 2 into a Markov model.
- Following the same steps as in task 2 for inference from a Markov network.

3.2 Inferences from Markov Network



```
+------+-----------+
| x1   |   phi(x1) |
+======+===========+
| x1_0 |    0.7549 |
+------+-----------+
| x1_1 |    0.0150 |
+------+-----------+
| x1_2 |    0.0605 |
+------+-----------+
| x1_3 |    0.1697 |
+------+-----------+
```

```
+------+-----------+
| x3   |   phi(x3) |
+======+===========+
| x3_0 |    0.1788 |
+------+-----------+
| x3_1 |    0.6622 |
+------+-----------+
| x3_2 |    0.1590 |
+------+-----------+
```

```
+------+-----------+
| x4   |   phi(x4) |
+======+===========+
| x4_0 |    0.6512 |
+------+-----------+
| x4_1 |    0.1523 |
+------+-----------+
| x4_2 |    0.0000 |
+------+-----------+
| x4_3 |    0.1965 |
+------+-----------+
```

```
+------+-----------+
| x5   |   phi(x5) |
+======+===========+
| x5_0 |    0.3763 |
+------+-----------+
| x5_1 |    0.1098 |
+------+-----------+
| x5_2 |    0.1062 |
+------+-----------+
| x5_3 |    0.4077 |
+------+-----------+
```

(a) (b) (c) (d)

Figure 10: Few Inferences from the Markov Model (a) marginal prob(x1) (b) p(x3|x2=0) (c) p(x4|x3=0,x2=4) (d) p(x5|x1=1)

3.4 Comparison between Inferences of Bayesian and Markov Network

3.4.1 Based on computation time

It was observed that the Markov Model was taking more time than the Bayesian Model for inference, although both of them took time in milliseconds. In Table 6 below, the exact computation times can be observed.

Table 6: Computation time taken for inference by Bayesian and Markov Model

Model	Inference Computation Time (ms)
Bayesian Model	24.039
Markov Model	24.063

4 Task 4

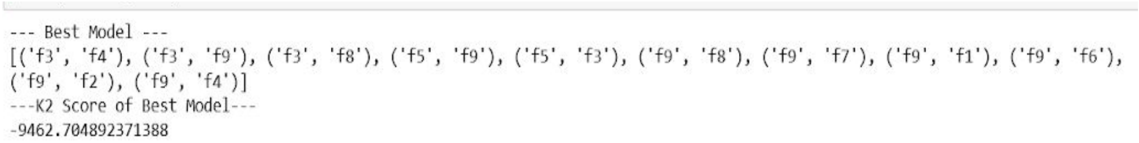
Use the "and" image dataset to construct a Bayesian network and evaluate the goodness score (likelihood of a dataset) of several Bayesian networks.

4.1 Steps Followed to Accomplish Task 4

- The given “AND” dataset was provided in a .csv file.
- This .csv file was converted to a dataframe using a python function.
- The obtained dataframe was passed to the K2Score function to get a K2Score object.
- This K2Score object and the dataframe was passed to the HillClimbSearch function of pgmpy. This function finds the best Bayesian model for that dataset based on their K2 scores.
- The edges of the obtained best model were passed to the BayesianModel function of pgmpy to create the Bayesian Model.
- This model was fit to the data using the fit() function of pgmpy library. This is done so that the model knows the probability distribution of the dataset.
- Once this is done get_cpds() function of pgmpy library was used to get the cpds.
- Using the Variable Elimination function, similar to task 2 and 3, inference was done to find the values of the 9 features given the values of others.
- Finally, the obtained bayesian model was converted to a markov model using the to_markov function in pgmpy library.

4.2 Results of Task 4

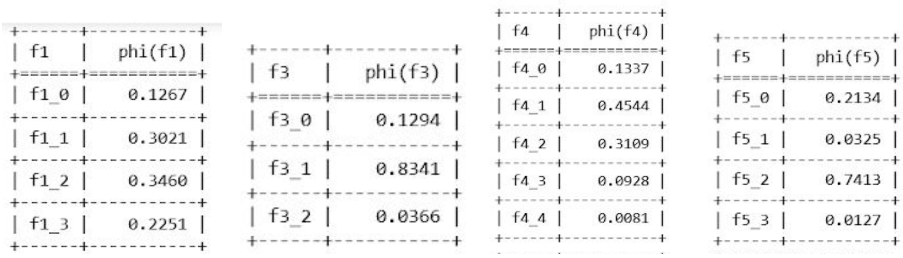
4.2.1 Obtained Best Bayesian Model and its K2 Score



```
--- Best Model ---
[('f3', 'f4'), ('f3', 'f9'), ('f3', 'f8'), ('f5', 'f9'), ('f5', 'f3'), ('f9', 'f8'), ('f9', 'f7'), ('f9', 'f1'), ('f9', 'f6'),
('f9', 'f2'), ('f9', 'f4')]
---K2 Score of Best Model---
-9462.704892371388
```

Figure 11: Best Model and its K2 Score

4.2.2 Inferences from the Best Bayesian Model



```
+------+-----------+
| f1   |   phi(f1) |
+======+===========+
| f1_0 |    0.1267 |
+------+-----------+
| f1_1 |    0.3021 |
+------+-----------+
| f1_2 |    0.3460 |
+------+-----------+
| f1_3 |    0.2251 |
+------+-----------+
```

(a)

```
+------+-----------+
| f3   |   phi(f3) |
+======+===========+
| f3_0 |    0.1294 |
+------+-----------+
| f3_1 |    0.8341 |
+------+-----------+
| f3_2 |    0.0366 |
+------+-----------+
```

(b)

```
+------+-----------+
| f4   |   phi(f4) |
+======+===========+
| f4_0 |    0.1337 |
+------+-----------+
| f4_1 |    0.4544 |
+------+-----------+
| f4_2 |    0.3109 |
+------+-----------+
| f4_3 |    0.0928 |
+------+-----------+
| f4_4 |    0.0081 |
+------+-----------+
```

(c)

```
+------+-----------+
| f5   |   phi(f5) |
+======+===========+
| f5_0 |    0.2134 |
+------+-----------+
| f5_1 |    0.0325 |
+------+-----------+
| f5_2 |    0.7413 |
+------+-----------+
| f5_3 |    0.0127 |
+------+-----------+
```

(d)

Figure 12: Few Inferences from the Bayesian Model (a) p(f1) (b) p(f1|f2=0) (c) p(f4|f3=0,f2=1,f5=0) (c) p(f5|f1=1)

5 Noteworthy Observations

- Although, randomly sampled data was used, the best “th” obtained did not change how many ever times I re-generated the dataset. This was because a Discrete Distribution Sampler function random.choice() was used to select the data. This selects the data based on the probability distribution. Hence, the data point with the highest probability is always selected the most number of times.
- Markov Model takes more time to infer as compared to Bayesian Network
- Although, the Bayesian Network formed using low cross entropy values was the best, which was expected, but the peculiar thing that was observed was that, comparing between two bayesian networks both of which were formed using lower cross entropy values, the one that included a few edges with slightly medium cross entropy edges was a better model than all edges with the least possible cross entropy.

Acknowledgments

I would like to thank Prof. Srihari for giving a project that helped understand the basics of Bayesian Networks. I would also like to thank T.As TieHang, Mihir and Muhammad for their guidance without whose insights completing this project would have been impossible

References

[1] https://piazza.com/class/jq8nawb0m0v7hy?cid=30

[2] TA: TieHang guidance