Advance Machine Learning CSE 674- Project 1

Apurba Mahanta
UB PERSON# **50288705**
University at Buffalo, The State University of New York
apurbama@buffalo.edu

Abstract

The goal of this project is to design probabilistic graphical models (PGMs) to determine probabilities of observations which are described by several features.For this project we took the handwriting patterns which are described by document examiners.The PGM's can then be used to determine whether a particular handwriting sample is common (high probability) or rare (low probability) and which in turn can be useful to determine whether a sample was written by a certain individual.We considered only the letter pair **th** in this study since it is the most commonly encountered pair of letters (called a bigram) in English.As a part of the project we aldo evaluated **and** image dataset to construct a Bayesian network and evaluate the goodness score (likelihood of a dataset) of several Bayesian networks constructed from it.

1 Feature definitions

1) x1 - Height Relationship of t to h
2) x2 - Shape of Loop of h
3) x3 - Shape of Arch of h
4) x4 - Height of Cross on t staff
5) x5 - Baseline of h
6) x6 - Shape of t

2 Evaluate pairwise correlations and independencies that exist in the data

To determine if x_i and x_j are independent we can check if $P(x_i, x_j)$ is almost equivalent to $P(x_i) \times P(x_j)$ where $P(x_i, x_j)$ is the joint probability distribution which is calculated as $P(x_i \mid x_j) \times P(x_j)$. We can measure this closeness by approximately calculating the value using the below formula.

$$\sum abs\left(\left(P(x,y) - P(x)P(y)\right)\right)$$

We got the following values of closeness between the features to decide the independent sets that might exists in the data.

1) P(x2 | x1) = 0.15977
2) P(x4 | x1) = 0.11943
3) P(x6 | x1) = 0.16015
4) P(x3 | x2) = 0.218525
5) P(x5 | x2) = 0.132460
6) P(x2 | x3) = 0.2187580

7) P(x5 | x3) = 0.1155199
8) P(x6 | x3) = 0.095640
9) P(x2 | x4) = 0.119570
10) P(x5 | x4) = 0.115699
11) P(x6 | x4) = 0.1434699
12) P(x2 | x5) = 0.1312649
13) P(x3 | x5) = 0.1167000
14) P(x1 | x6) = 0.1603699
15) P(x2 | x6) = 0.1753150
16) P(x3 | x6) = 0.1390300
17) P(x4 | x6) = 0.1430700

From the above values we can get the independent sets by setting the significance level at 0.12.We consider the pairs whose closeness value is below 0.12 as the independent sets.

3 Construct various Bayesian network with the fewest number of edges that maximizes the likelihood.

3.1 Threshold

We set the threshold or significance level at **0.12** on the previous calculated result to determine if two variables are independent .

3.2 Bayesian Model Creation

1)Based on the above results we generated 5 Bayesian models with the minimum edges possible and the conditional probability distributions given to us for each node.To determine the goodness of the models we used K2 score method provided by PGMPY library.The pre requisite to get the K2score is to generate the sample data .

2)The sample data size of 5000 was generated where 1000 data size each was generated by respective models and then combined and shuffled .This sample data is then used in K2score method to determine the best model .

3)Ancestral sampling is used for data generation which is implemented by the class pgmpy.sampling. BayesianModelSampling.This class is used to generate the data for the respective models.

3.2.1 Bayesian Network Model-1

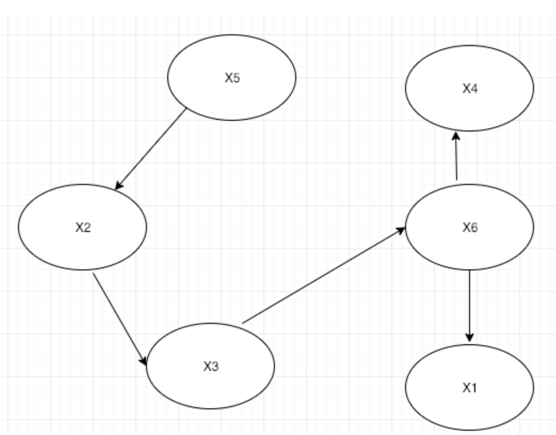

Figure 1: Bayesian Model 1

3.2.2 Bayesian Network Model-2

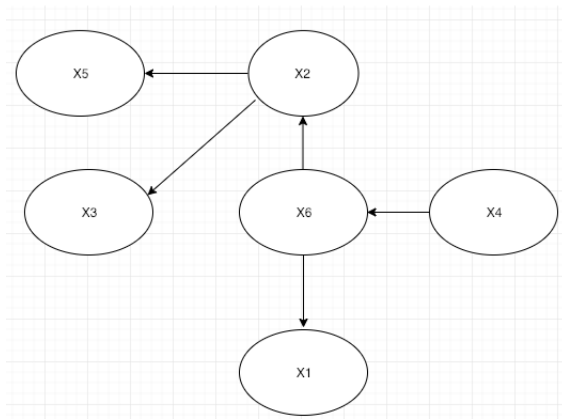

Figure 2: Bayesian Model 2

3.2.3 Bayesian Network Model-3

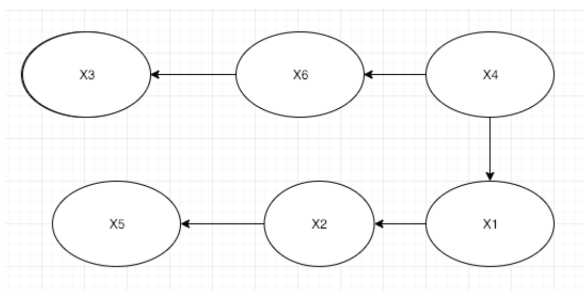

Figure 3: Bayesian Model 3

3.2.4 Bayesian Network Model-4

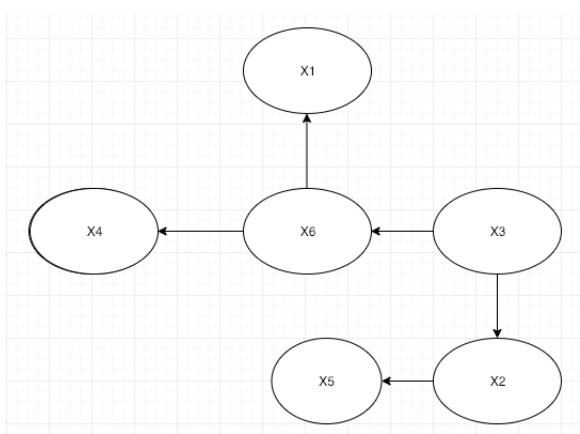

Figure 4: Bayesian Model 4

3.2.5 Bayesian Network Model-5

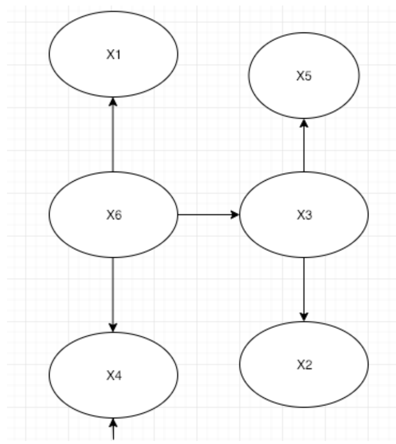

Figure 5: Bayesian Model 5

3.2.6 Model Performance Evaluation based on K2 Score

Bayesian Model 1 K2 Score	**-31762.55432**
Bayesian Model 2 K2 Score	**-31785.13432**
Bayesian Model 3 K2 Score	**-32090.000980**
Bayesian Model 4 K2 Score	**-31761.52891**
Bayesian Model 5 K2 Score	**-31819.69284**

Based on the above results we can say that Bayesian Model 4 performs the best among all of them.

3.2.7 Observations

1) High Probability TH :

X1	**X2**	**X3**	**X4**	**X5**	**X6**
0	**1**	**1**	**0**	**3**	**3**

2)Low Probability TH:

X1	**X2**	**X3**	**X4**	**X5**	**X6**
2	**4**	**0**	**1**	**0**	**4**

3) If there is a V structure being formed ,there was insufficient data to proceed for as for example $P(X_6 \mid P(X_1, X_4))$. We did not have the conditional probability distribution given 3 features.However as per the D separation we can rewrite $P(X_6 \mid P(X_1, X_4))$ as $P(X_1 \mid X_6) \times P(X_4 \mid X_6) \times P(X_6) \mid P(X_1, X_4)$. However it was observed that the conditional probability distribution did not sum up to 1 and was more than that,hence was not able to proceed with that particular Bayesian Network model.

4 Bayesian network into a Markov network using moralization.

1) Markov network is generated from the the best Bayesian Network model using the method to_markov_model() given in the pgmpy library.

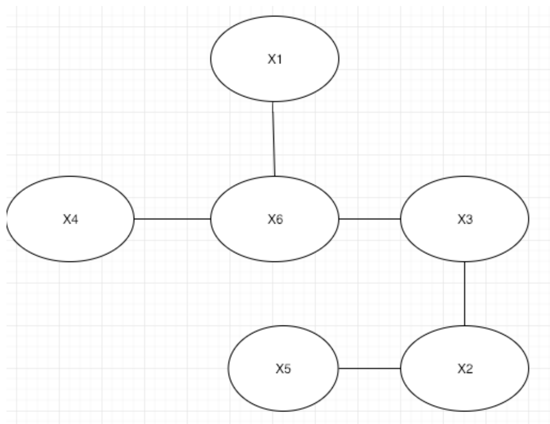

Figure 6: Markov Model

4.1 Inferences using Bayesian network and the Markov network

1) In inference we try to answer probability queries over the network given some other variables. For example we might want to know the probable values of Shape of Arch of h given the Shape of Loop of h.Variable Elimination library from pgmpy was used for inference for both the Bayesian and the Markov network model.The example below depicts the inference process in both bayesian and markov networks.

```
infer = VariableElimination(writerModel4)
print(infer.query(['x3'], evidence={'x5': 3}) ['x3'])
print(infer.query(['x2'], evidence={'x5': 3}) ['x2'])
print(infer.query(['x1'], evidence={'x5': 3}) ['x1'])
print(infer.query(['x4'], evidence={'x5': 3}) ['x4'])
print(infer.query(['x6'], evidence={'x5': 3}) ['x6'])
```

```
+------+-----------+
| x3   |   phi(x3) |
+======+===========+
| x3_0 |    0.1778 |
+------+-----------+
| x3_1 |    0.6475 |
+------+-----------+
| x3_2 |    0.1746 |
+------+-----------+
+------+-----------+
| x2   |   phi(x2) |
+======+===========+
| x2_0 |    0.2684 |
+------+-----------+
| x2_1 |    0.3291 |
+------+-----------+
| x2_2 |    0.0000 |
+------+-----------+
| x2_3 |    0.1465 |
+------+-----------+
| x2_4 |    0.2560 |
+------+-----------+
```

Figure 7: Inferences by Bayesian Model

```
infer_markov = VariableElimination(markov)
print(infer_markov.query(['x3'], evidence={'x5': 3}) ['x3'])
print(infer_markov.query(['x2'], evidence={'x5': 3}) ['x2'])
print(infer_markov.query(['x1'], evidence={'x5': 3}) ['x1'])
print(infer_markov.query(['x4'], evidence={'x5': 3}) ['x4'])
print(infer_markov.query(['x6'], evidence={'x5': 3}) ['x6'])
```

```
+------+-----------+
| x3   |   phi(x3) |
+======+===========+
| x3_0 |    0.1778 |
+------+-----------+
| x3_1 |    0.6475 |
+------+-----------+
| x3_2 |    0.1746 |
+------+-----------+
+------+-----------+
| x2   |   phi(x2) |
+======+===========+
| x2_0 |    0.2684 |
+------+-----------+
| x2_1 |    0.3291 |
+------+-----------+
| x2_2 |    0.0000 |
+------+-----------+
| x2_3 |    0.1465 |
+------+-----------+
| x2_4 |    0.2560 |
+------+-----------+
```

Figure 8: Inferences by Markov Model

1) Hence we can say that the inferences provided by the markov and bayesian models are very much accurate based on the data provided to us.

2) The computation time for bayesian and markov networks are quite similar but the markov network models are somewhat on the higher side than bayesian,although the difference is not much.

Bayesian Total Query Time for 100 queries in sec :1.05370900
Markov Total Query Time for 100 queries in sec :1.4753679

5 Construction of Bayesian and Markov networks based on the "and" image dataset

1) The data of 9 features of the and dataset was given to us.

2) To estimate the best model given the data ,we used HillClimbSearch technique that is available at from pgmpy.estimators.HillClimbSearch implements a greedy local search that starts from the DAG start (default: disconnected DAG) and proceeds by iteratively performing single-edge manipulations that maximally increases the score. The search terminates once a local maximum is found.Exaustive Search is another technique to get the best model given data ,but it is usually not recommended when the number of feature is more than 5 which is 9 in our case;
3)The best model generated by HillClimbSearch along with the K2 score is as below:

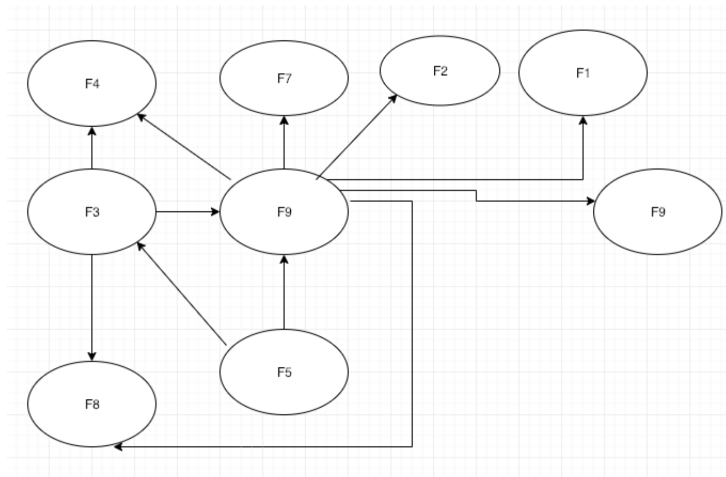

Figure 9: AND Features Bayesian Model by HillClimb Search

3) To try out different models , conditional independence is tested out using ConstraintBasedEstimator class and test_conditional_independence method between various features.We got the following results from it by keeping the significance level at 0.005 :

is_independent('f3', 'f4') - False
i_independent('f3', 'f9')) - True
is_independent('f3', 'f8')) - False
is_independent('f5', 'f9')) - False
is_independent('f5', 'f3')) - False
is_independent('f9', 'f8')) - False
is_independent('f9', 'f7')) - False
is_independent('f9', 'f1')) - False
is_independent('f9', 'f6')) - False
is_independent('f9', 'f2')) - True
is_independent('f9', 'f4')) - True

4) We constructed 4 more bayesian networks using the above independence relations:

Bayesian Model 1 :
('f3', 'f4'), ('f3', 'f8'), ('f5', 'f9'), ('f5', 'f3'), ('f9', 'f8'), ('f9', 'f7'), ('f9', 'f1'), ('f9', 'f6'), ('f9', 'f2')

Bayesian Model 2 :
('f3', 'f4'), ('f5', 'f9'), ('f5', 'f3'), ('f9', 'f8'), ('f9', 'f7'), ('f9', 'f1'), ('f9', 'f6')

Bayesian Model 3 :
('f4', 'f3'), ('f9', 'f5'), ('f3', 'f5'), ('f8', 'f9'), ('f7', 'f9'), ('f1', 'f9'), ('f6', 'f9')

Bayesian Model 4 :
('f4', 'f3'), ('f3', 'f8'), ('f5', 'f9'), ('f5', 'f3'), ('f9', 'f8'), ('f9', 'f7'), ('f9', 'f1'), ('f9', 'f6'), ('f2', 'f9')

5.1 K2 Score Evaluation

AND Model	K2 Score
Hill Climb Search Model	**-9462.7048**
Bayesian Model 1	**-9472.1900**
Bayesian Model 2	**-8170.1712**
Bayesian Model 3	**-8332.6588**
Bayesian Model 4	**-9504.2481**

5.2 Observations

1) If we consider that (f2 , f9) and (f4 , f9) features are independent based on the significance level the k2 score improves to -8170.1772.

2)But we cannot conclude this as the data might be less to determine the correlation between the features.

3)Hence we consider Hill Climb Search model as the best model among them.

5.3 Markov Model Generation

1) Markov model is generated from the best bayesian model using moralization method to_markov_model() from the pgmpy library.

5.4 Conditional Probability Generation

1) We generated the condition probability distributions for the best model using MaximumLikelihoodEstimator and BayesianEstimator. Slight difference was observed between the 2 methods as BayesianEstimator uses K2 prior, which simply adds 1 to the count of every single state and then later updated using the state counts from the observed data.

2)The approach in Maximum Likelihood Estimation (MLE) is to fill the CPDs in such a way, that $P(data \mid model)$ is maximal and uses relative frequencies.

5.5 Inference

1) Variable Elimination method is used to generate inference given certain evidence of the features.Similar results was observed for both bayesian and markov model.

```
from pgmpy.inference import VariableElimination
infer = VariableElimination(best_model)
print(infer.query(['f1'], evidence={'f9': 0}) ['f1'])
```

```
+------+-----------+
| f1   |   phi(f1) |
+======+===========+
| f1_0 |    0.1991 |
+------+-----------+
| f1_1 |    0.4459 |
+------+-----------+
| f1_2 |    0.2641 |
+------+-----------+
| f1_3 |    0.0909 |
+------+-----------+
```

Figure 10: Inferences example for the **and** model

5.6 Likelihood of the dataset

1)For both the markov and bayesian models ,we can verify that the inference produced by the models matches with the estimated conditional probability distributions as shown below.

f9	f9(1)	f9(2)
f1(0)	0.19913419913419914	0.10709838107098381
f1(1)	0.4458874458874459	0.2602739726027397
f1(2)	0.26406926406926406	0.3686176836861768
f1(3)	0.09090909090909091	0.2640099626400996

Figure 11: CPD for P(F1 | F9)

f1	phi(f1)
f1_0	0.1991
f1_1	0.4459
f1_2	0.2641
f1_3	0.0909

Figure 12: Inferences query for F1 given F9 as 0

2)Hence we can conclude that dataset of "and" fetaures is quite relevant and produces inference as per the expectation.

3)The computation time for both the bayesian and markov model is also very similar.

4) The computation time for both the bayesian and markov models are similar with the markov model being slightly on the higher side.
Bayesian And Model Total Query Time for 100 queries in sec :2.232877999999
Markov And Model Total Query Time for 100 queries in sec :2.516594999999999

References

[1] http://pgmpy.org/