Determining Probabilities of Handwriting Formations using PGMs

Parth Shah
SEAS, University at Buffalo
Buffalo, NY, 41214
Person Number: 50291125
parthnay@buffalo.edu

Abstract

The project aims to develop a probabilistic graphical model (PGMs) to determine the probabilities of observations of handwriting patterns described by document examiners.

1 Introduction

The given problem is an inference problem. We use PGM to recognize the patterns in the bigram '*th*' in samples provided by multiple observers and determine the high probable patterns (most occurrences) and the low probable pattern (least occurrences). So as to make a prediction of which of the writer has written the new sample. The project is divided into 3 tasks

- ➔ Task 1: Determining Dependencies
- ➔ Task 2: Bayesian Model Creation and Inference
- ➔ Task 3: Bayesian To Markov Model and Inference of Markov Model
- ➔ Task 4: Bayesian and Markov Models for AND Dataset

2 DataSet and Data PreProcessing

We are provided with Marginal Probabilities for all 6 features of the dataset namely - x1, x2,x3,x4,x5,x6 in Table2. Along with the Marginal Probabilities we were provided with Conditional Probability Distributions(CPD) of variables with respect to others in Tables 3 to 8. The Tables provided contained values of CPD in percentage and number of observers that provided that sample.
We were also provided with an AND-Dataset which contained variable features for samples provided by AND writers.

3 Task 1: Determining Dependencies

We were tasked with finding the dependencies in the attributes for the data '*th*'. And to Evaluate pairwise correlations and independence that exist in the data. We used cross-entropy to calculate pairwise corelations between data given their CPD.

The cross-entropy between two probability distributions **p** and **q** over the same underlying set of events measures the average number of bits needed to identify an event drawn from the set.[1] If **p(a,b) = p(a) * p(b)** then a and b are said to be independent. Where **p(a,b) = p(a/b)p(b)**

Given the CPD's perfectly matching distribution cannot be attained always thus we set a threshold value. The pairwise correlation above the threshold are dependent and the ones below are independent. The results for Task 1 can be shown as

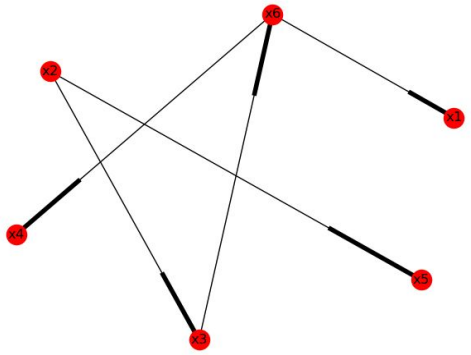
```
Considering threshold as 0.14
Dependent Nodes are :
x2 and x1 are dependent
x6 and x1 are dependent
x3 and x2 are dependent
x2 and x3 are dependent
x6 and x4 are dependent
x2 and x5 are dependent
x1 and x6 are dependent
x2 and x6 are dependent
x4 and x6 are dependent
```

```
[['x1', 'x2'], ['x1', 'x6'], ['x2', 'x3'], ['x3', 'x2'], ['x4', 'x6'], ['x5', 'x2'], ['x6', 'x1'], ['x6', 'x2'], ['x6', 'x4']]
```

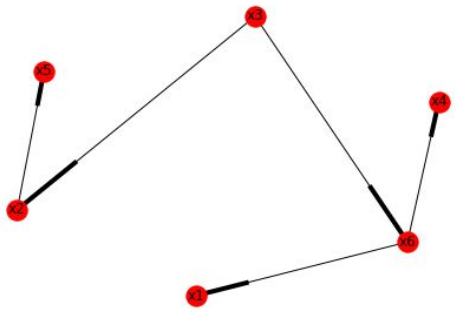
4 Task 2: Bayesian Model Creation and Inference

Bayesian Model Creation

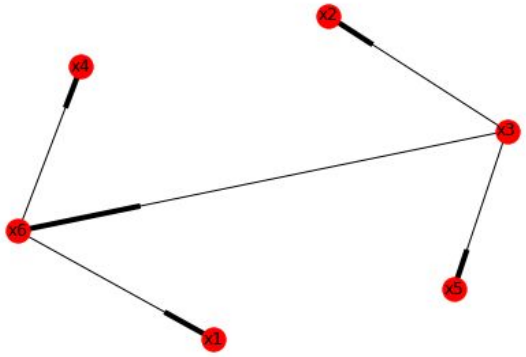
From the dependencies determined above in Task 1, we created Bayesian models based on them and tested the models against each other using K2Score as metric and data generated using Bayesian Sampling on each model as the data for each model. The Models were generated manually depending upon the independencies in the pairs.The Models thus created were



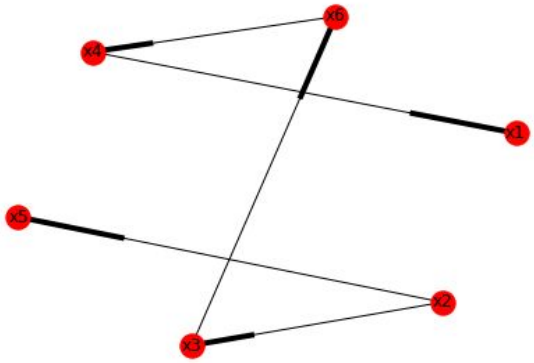
Model1



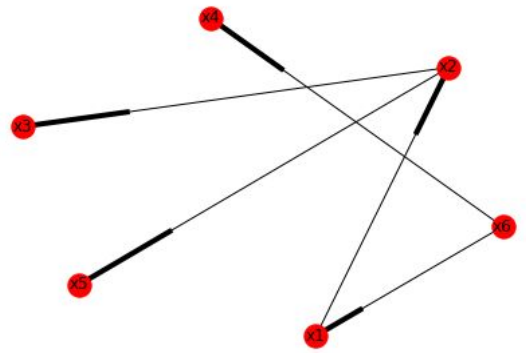
Model2



Model3



Model4

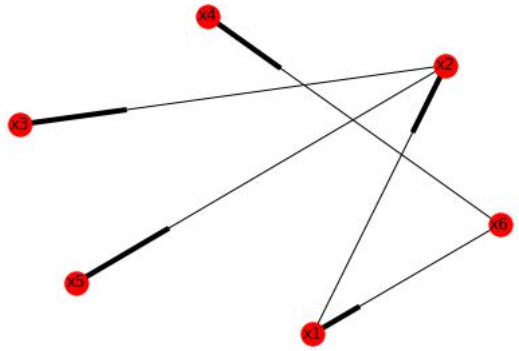


Model5

Inference

Given the Bayesian Models, we used K2 score as a metric to compare two models. We used **class pgmpy.sampling.Sampling.BayesianModelSampling(model)** to generate samples for each model. Using this data we generate a K2Score for each model and the model with the highest K2Score is selected as the best model.

With the best model Selected we then used the data from that model to infer repeating patterns for high probability and low probability '*th*' data and describe the found data. We found the best model to be



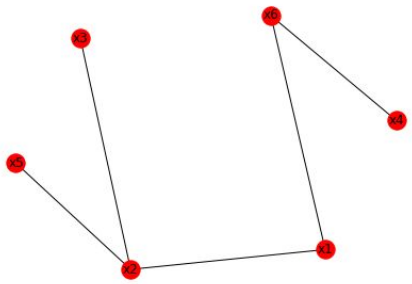
Model5

The Inferred result can be shown as

```
Model 1 K2 Score: -6483.121723396625
Model 2 K2 Score: -6482.915778986373
Model 3 K2 Score: -6503.677382018984
Model 4 K2 Score: -6487.01020922778
Model 5 K2 Score: -6448.446045605671

The Best Model is : Model 5
The Worst Model is : Model 1

For Best Bayesian Model
High Probability th characteristics are :
Height Relationship of t to h : t shorter than h
Shape of Loop of h : curved right side and straight left side
Shape of Arch of h : pointed
Height of Cross on t staff : upper half of staff
Baseline of h : no set pattern
Shape of t : closed

Low Probability th characteristics are :
Height Relationship of t to h : t shorter than h
Shape of Loop of h : retraced
Shape of Arch of h : pointed
Height of Cross on t staff : upper half of staff
Baseline of h : no set pattern
Shape of t : tented
```

5 Task 3: Bayesian To Markov Model and Inference of Markov Model

For Task 3 We are required to convert the best Bayesian Model to a Markov Model.
We use moralisation to convert the Bayesian Model to a Markov Model and then use the inference as given above to generate a pattern for high and low probable '*th*' patterns. We use class **pgmpy.sampling.Sampling.GibbsSampling(model=None)[source]** to generate data for this Markov Model.
The Markov model thus produced for the best models is



Markov Model for Best Model

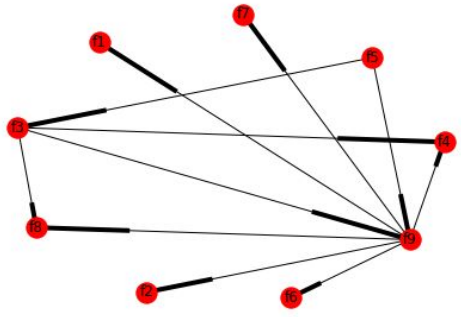
The Inferred results can be written as

```
For Markov Model
High Probability th characteristics are :
Height Relationship of t to h : t shorter than h
Shape of Loop of h : curved right side and straight left side
Shape of Arch of h : pointed
Height of Cross on t staff : upper half of staff
Baseline of h : no set pattern
Shape of t : closed

Low Probability th characteristics are :
Height Relationship of t to h : t shorter than h
Shape of Loop of h : retraced
Shape of Arch of h : pointed
Height of Cross on t staff : upper half of staff
Baseline of h : no set pattern
Shape of t : tented
```

Thus we can see that the inferences drawn from the Markov model match the ones from the Bayesian Model.

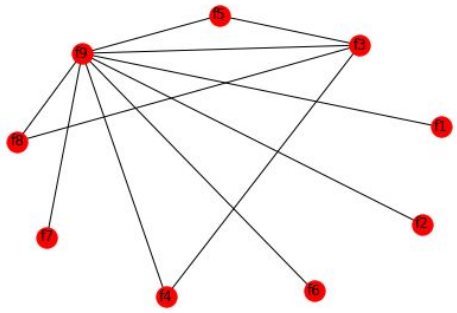
6 Task 4: Bayesian and Markov Models for AND Dataset

We were provided with an AND Dataset which contained samples for '*and*' as written by different writers. We used class pgmpy.estimators.HillClimbSearch.HillClimbSearch(data, scoring_method=None, **kwargs) to get the best Edges and then construct a Bayesian model with those edges. We use K2 Score as a scoring method to find out the best edges from the data in hill climbing. The Bayesian model thus created can be seen below



Bayesian AND Model

Then we convert this Bayesian Model to Markov Model as done in task 3. The Markov model thus created was as below



Markov AND Model

7 Conclusion

We used pgmpy to create a Bayesian model given CPD's and picked the best model using K2Score as a metric. We then used pgmpy to convert these Bayesian models into Markov Models and drew inferences from both and found them to be matching. We then used a given Dataset to create a Bayesian model using Hill Climbing technique and finally converted that model to Markov Model as well.

Acknowledgements

This report was prepared in accordance with the Project1(1).pdf Project_tut_v1.pdf and the pdf of the book provided, pgmpy documentation, cross entropy and PGM definitions available on Wikipedia.

References

[1] Probabilistic Graphical Models: Principles and Techniques Book by Daphne Koller and Nir Friedman

[2] Cross Entropy - https://en.wikipedia.org/wiki/Cross_entropy

[3] pgmpy - http://pgmpy.org/index.html

[4] https://medium.com/@Dezhic/understanding-probabilistic-graphical-models-658b6fa40184