Project 1: Determining Probabilities of Handwriting Formations using PGMs

Koustubh Vijay Kulkarni[*]
Department of Computer Science
University at Buffalo
Buffalo, NY 14214
kkulkarn@buffalo.edu

Abstract

The main objective of the project is to develop probabilistic graphical models (PGMs) to determine probabilities of observations which are described by various variables. We have given characterization of the structure of letter pair 'th', which has six random variables x1-x6 which take a set of discrete values. They can 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.

1 Tasks

This project is divided into 4 tasks. We have given with the conditional probability distribution tables of all the six variables and from that we need to generate various DAGs and then we need to evaluate them and choose the best amongst them.
But important task before generating the Bayesian Network is to understand the inter-dependency between the variables.

2 Task 1:-Evaluate Correlation and Independence Among the Variables

In this task, we need to find whether a particular variable is dependent on another. We can determine whether Xi and Xj are independent by testing if p(Xi,Xj) - p(Xi)p(Xj) is equal or nearly equal to 0. If the difference between joint probability and p(Xi)p(Xj) is close to 1, then that indicates they are highly correlated. On the other hand if the difference is close to 0, then we can say that they are independent. As we are given with the conditional probability with us, we can easily calculate joint probability of each pair of variables and then perform this operation to get the closeness. I have used below method to approximately calculate closeness:

$$\sum_1^n abs(p(Xi,Xj) - p(Xi)p(Xj))$$

Here is the table of each variable pair and the closeness measure:-

[*] Use footnote for providing further information about author (webpage, alternative address)—*not* for acknowledging funding agencies.

32nd Conference on Neural Information Processing Systems (NIPS 2018), Montréal, Canada.

Variable-Pair	Value
X3-X2	0.21875800000000004
X2-X3	0.21852500000000005
X6-X2	0.175315
X6-X1	0.16036999999999998
X1-X6	0.16015500000000005
X1-X2	0.15977000000000002
X4-X6	0.14346999999999996
X6-X4	0.14307
X2-X5	0.13481000000000007
X5-X2	0.131522
X4-X1	0.11957000000000005
X1-X4	0.11943000000000004
X3-X6	0.11768
X4-X2	0.11569999999999997
X3-X5	0.11391999999999995
X5-X3	0.11362000000000003
X6-X3	0.09434000000000006

In order to determine which pairs of the variables to include as edge in the graph we need to select such variables which are highly correlated on each other. Here we need to apply a certain **threshold** in order to select the pairs which have value more than that of threshold.

3 Task 2: Construction of Bayesian Network

Once we get the closeness value of each pair of variable we need to construct several DAGs. The approach I used is, I set certain value of **threshold**, and considered minimum number of edges that cover most of the nodes. I gradually increased threshold to get different graphs. Maximum value of the threshold which includes all the 6 variables as node in the graph is **0.13**. I generated total 10 DAGs.

Model No.	Edges
M1	[(’x4’, ’x6’), (’x6’, ’x2’), (’x6’, ’x3’), (’x3’, ’x5’)]
M2	[(’x1’,’x4’),(’x1’,’x2’),(’x2’,’x3’),(’x3’,’x5’),(’x4’,’x6’)]
M3	[(’x6’,’x4’),(’x6’,’x2’),(’x6’,’x1’),(’x2’,’x3’),(’x2’,’x5’)]
M4	[(’x6’,’x4’),(’x6’,’x2’),(’x2’,’x3’),(’x2’,’x5’),(’x4’,’x1’)]
M5	[(’x6’,’x4’),(’x6’,’x1’),(’x1’,’x2’),(’x2’,’x5’),(’x2’,’x3’)]
M6	[(’x4’, ’x6’), (’x6’, ’x2’), (’x6’, ’x3’), (’x3’, ’x5’)]
M7	[(’x4’,’x6’),(’x6’,’x2’),(’x6’,’x1’),(’x2’,’x3’)]
M8	[(’x1’,’x2’),(’x1’,’x4’),(’x2’,’x3’),(’x3’,’x5’)]
M9	[(’x6’,’x2’),(’x6’,’x1’),(’x2’,’x3’)]
M10	[(’x6’,’x2’),(’x2’,’x3’)]

3.1 Evaluate the Best Model

In order to evaluate best model, I decided to use K2Score Method. K2 Score is used for Bayesian structure scoring for Bayesian Models with Dirichlet priors.It measures how well a model is able to describe the given data set. But in our case, we do not have data with us. We only have CPDs of various nodes. In order to generate the data from these CPDs I have used **Forward Sampling**.

3.1.1 Various Approaches to Determine best Model with K2 Score

1. **Evaluate K2 Score from Individual Sampling**
   In order to get the K2 score of the model, we need to generate the data by using sampling methods. Once we get the data, we pass the data to scoring function and then it will give us the score. Higher the score better the model.
   But one problem with such approach is, the data that we get from sampling of the one model is biased towards that model. Moreover, for each model we will generate new data from sampling. So each model will be evaluated on different datasets. Then question comes, **how**

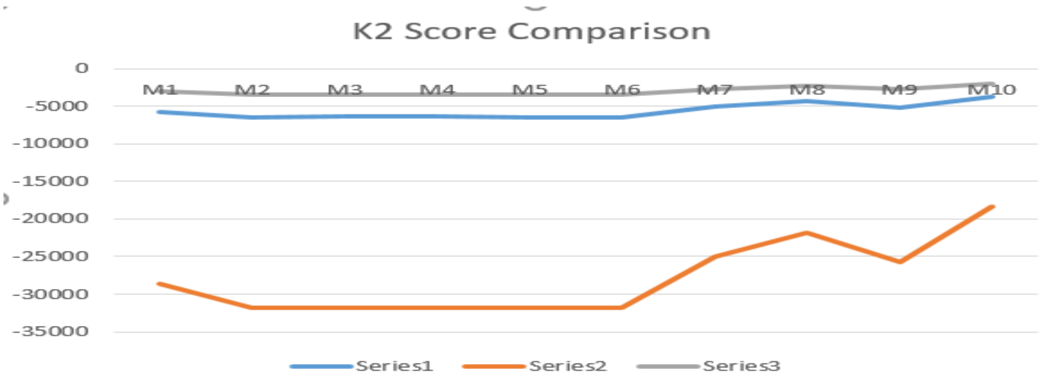
can we compare K2 scores of models which are evaluated on different datasets?

2. **Evaluate K2 Score on All the Collected Samples**
   In order to deal with above mentioned situation I calculated K2 score of each model separately. Stored the data generated from each of the model and combined the whole data. So, I separately generated 1000 records for each of the model and then combined all of them to get one big data set of 10000 records. Now I evaluated each of the 10 models for against this data and got the K2 Score of each model. Here, as the data is same for all the models, we can say all the models are now **fairly evaluated**.

3. **Evaluate K2 Score on Shuffled subset of All the Collected Samples**
   Another variation that I tried is, I shuffled this 10000 records and randomly took 1000 records out of it and then I measured K2 Score for each model.

3.1.2 Comparison of K2 scores of all the models

Model	Indv. Sampling(1k)	Combined Samples(10k)	Subset of Combined Samples(1k)
M1	-5698.615008662573	-28646.467290536348	-3049.5133412398154
M2	-6458.183007102665	-31811.94209963729	-3372.112313459687
M3	-6350.856251620129	-31783.30097149976	-3391.188343388795
M4	-6378.828094732967	-31771.0542877177	-3390.4981240595916
M5	-6445.621482566455	-31763.65587961477	-3382.771976740407
M6	-6406.069484785381	-31755.77376068471	-3374.8655898964907
M7	-5037.59351209235	-24948.624979932727	-2633.495363397922
M8	-4356.609364365477	-21790.037755829035	-2296.3051271045933
M9	-5202.17055070369	-25776.570398568154	-2717.370143549516
M10	-3673.7152343061534	-18321.324518616082	-1940.3093767921168



If we take a look at these results, model no. 10 i.e. **M10** has best K2 score for all the variations that I tried. On the basis of these results and after trying out these 3 approaches to evaluate K2 Score, I conclude that out of the 10 models that I tried, M10 - **(x6->x2->x3)** is the best Bayesian Model.

3.2 What Best Model Tells Us?

If we look at the closeness score of the all edges in the model M10, we will find out that they are among the top 3 edges which are highly correlated with each other. Other edges(e.g. X6-X3, X5-X3 etc.) have very low closeness values indicating they are independent. Hence, the model that I got performed better and got good K2 score as it has highly dependent nodes showing great causal relationship.
It indicates variable X2 (Shape of Loop of h) is dependent on X6 (Shape of t) and X3 (Shape of Arch of h) is dependent on X2 (Shape of Loop of h).

3.3 Determining High Probability ’th’

Now as we got our best model, we need to determine how high probability ’th’ looks like. I have tackled this problem in twp ways:-

1. **Calculating unique combination of variables in data set**:
   This is the most easy way to find out which pattern of the data occurs most in the data set and that data will represent the high probability ’th’. In order to do that I counted highest number of unique repeating pattern in the combined data set. There were total 95 such repeating records in the dataset of 10000 and found out below values for such ’th’:-

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

2. **Using MAP query to determine most likely values**:
   I also used this approach to find out most likely values of the other variables given values of ’x6’. I repeated this process for all values of ’x6’ and got all possible combination of values of other variables from MAP query.

   ```
   G = BayesianModel([('x6','x2'),('x2','x3')])
   G.add_node('x4')
   G.add_node('x5')
   G.add_node('x1')
   G.fit(pd.read_csv('AllSampleData.csv'))
   infr=VariableElimination(G)
   qer =infr.map_query(['x3','x2','x4','x5','x1'],evidence={'x6':0})
   print(qer['x5'])
   print(qer['x4'])
   print(qer['x3'])
   print(qer['x2'])
   print(qer['x1'])
   ```

   Now, I took majority value for each of the variable and considered that as high probability ’th’. The results I got are same as the one I got from first approach.

	x1	**x2**	**x3**	**x4**	**x5**
0	0	3	1	0	3
1	0	0	1	0	3
2	0	1	1	0	3
3	0	1	1	0	3
4	0	1	1	0	3

How it looks?
So, if we compare the results of both the ways we get following **’th’** as high probability ’th’: It has **t shorter than h**, shape of loop of h is **curved right side and straight left side**, shape of arch of h is **pointed**, Height of cross on t staff is **upper half of staff**, Baseline of h has **no set pattern** and shape of t is **closed**.

3.4 Determining Low Probability ’th’

I applied similar approach to determine the low probability of ’th’. Instead of counting highest number of repeating pattern I counted minimum number of repeating rows. Below are the values of all the variables representing low probability ’th’:-

x1	x2	x3	x4	x5	x6
0	0	0	0	2	3

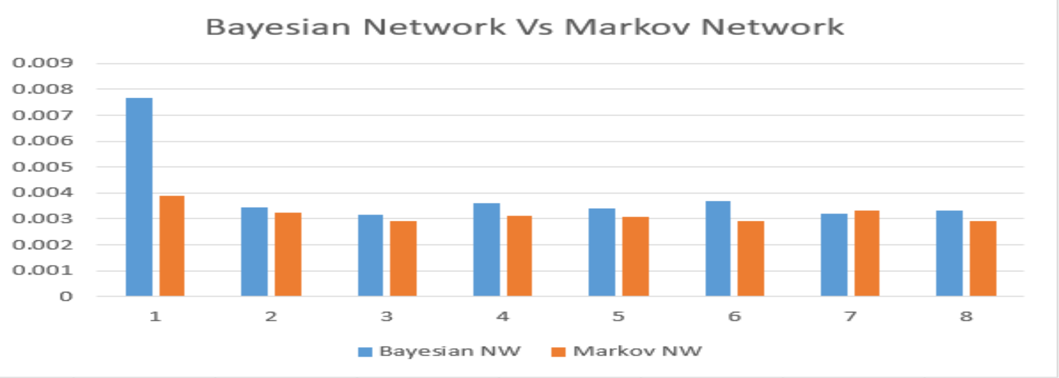
How it looks?
So, we get following **’th’** as low probability ’th’: It has **t shorter than h**, shape of loop of h is **retracted**, shape of arch of h is **rounded arch**, Height of cross on t staff is **upper half of staff**, Baseline of h has **baseline even** and shape of t is **closed**.

4 Task 3: Conversion of Best Bayesian Network to Markov Network

Moralization is the process in which directed graph of Bayesian network is converted into undirected graph. In moralization, if there is directed edge present between the two nodes then that is converted into an undirected edge. Also, if the two nodes are parents of the same node then there exists an edge between them.
In PGMPY, there is a function (**to_markov_model**()) to convert Bayesian Network into Markov Network. So, I converted my best Bayesian Model:- **(x6->x2->x3)** into Markov model .

4.1 Comparison inferences in terms of Time

In order to compare the performance of both the models I queried same query on both the models and read the time it took to give the inferences. I ran several different queries and timed the performance of both the models.



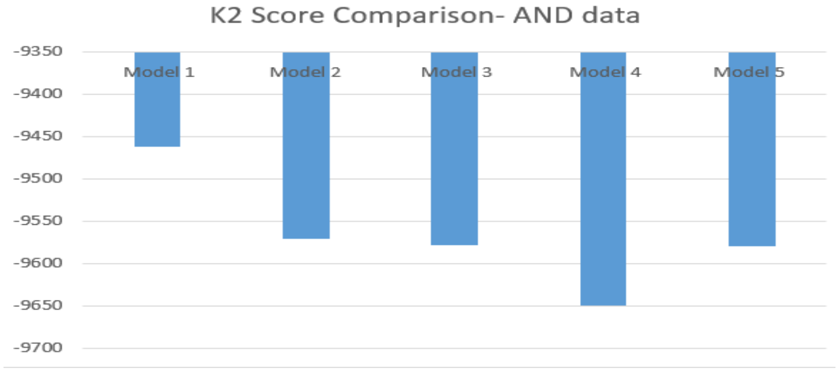
Here, by taking a quick look at the charts we can infer that overall, Markov Network performs faster than the Bayesian Network.

5 Task 4: Bayesian Network For AND image Dataset

In this task 4, we have to create Bayesian Network for AND image dataset. Unlike the ’th’ dataset, we have the data for the AND datset. We need to identify the best model as per the given data. There are various methods to identify the structure of model based on the data. One of such method is **Exhaustive Search**. This method computes all possible directed acyclic graphs with a given set of nodes, but drawback of this method is it only works for nodes<6. In our case of ’AND’ data set, there are total 9 features. So we can’t use this method for determining the structure of the model.

Another popular method is, **Hill-Climb search**. This method Performs local hill climb search to estimates the Bayesian Model structure that has optimal score, according to the scoring method supplied. I supplied ’K2Score’ scoring mechanism, alternatively we can use other scoring methods such as ’BicScore’.
Hill-Climb search will give us the best possible structure of the graph. Here is the edges returned by the Hill-climb search for AND data set:- [(’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 this model- -9462.704892371386

Model No.	K2 Score
Model 1	-9462.704892371386
Model 2	-9571.119526729564
Model 3	-9578.734130607785
Model 4	-9649.465162804985
Model 5	-9579.781145953315



Here, one thing to observe is, the scores that we got for other models are lower than the score we got from the best model.

6 Final Evaluation

I will summarize my results in this section. Our objective of this project is to learn about the Bayesian Networks. With the help of the CPDs associated with variables of 'th' dataset we built several Bayesian Networks and found out the best one based on K2 scores. I got the best model as-**(x6->x2->x3)**.

We identified high and low probability 'th' representation from the data set and converted best bayesian network into markov network and compared the inferences against time taken.
Finally we generated several DAGs for 'AND' dataset and identified best one using Hill-Climb algorithm.

References

[1] Class and Recitation notes.

[2] https://stackoverflow.com/questions/39964558/pandas-max-value-index.

[3] http://pgmpy.org/models.html.

[4] https://github.com/pgmpy/.

[5] https://stackoverflow.com/questions/35584085/how-to-count-duplicate-rows-in-pandas-dataframe.

[6] http://www.lx.it.pt/ asmc/pub/talks/09-TA/ta_pres.pdf

[7] http://www.lx.it.pt/ asmc/pub/talks/09-TA/ta_pres.pdf