Report of CSE 674 Project1: Determining Probabilities of Handwriting formation using PGM

Ankit Kumar Sinha*
UBIT name:ankitsin
Person: 50286874
ankitsin@buffalo.edu

Abstract

In this project, we had to create a Probabilistic Graphical Model(PGMs) to determine whether a handwriting sample is common (high probability) or rare(low probability). The sample used in this consisted of 'th' pair. This helped in identifying whether a sample was written by a certain individual.

1 Task:

1. Evaluate pairwise correlation and independencies in the dataset.
2. Construct a Bayesian Network with the fewest number of edges that maximizes the likelihood. Also, describe the high probability 'th' and low probability 'th' looks like.
3. Convert the Bayesian model into the Markov model using moralization. Compare the Bayesian model and the Markov model in terms of computation time.
4. Use the "And" dataset and calculate the goodness score of several Bayesian networks.

2 Task1 :

Steps followed for the task are as following:

1. We were provided with data of marginal probability and joint probability. These data were pre-processed to extract the required probabilities.
2. Using the equation,

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

the closeness of P(x,y) and P(x)P(y) was calculated. This gave us the correlation (independencies) of each feature.
The correlations between all the features are shown in the table:

X1	X2	X3	X4	X5	X6
0.	0.	0.	0.11957	0.	0.16037
0.15977	0.	0.218758	0.1157	0.12939	0.175315
0.	0.218525	0.	0.	0.115965	0.094025
0.11943	0.	0.	0.	0.	0.14307
0.	0.12926	0.11552	0.	0.	0.
0.160155	0.	0.11258	0.14347	0.	0.

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

31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.

3 Task 2:

In this task, I had to construct different Bayesian Model and find the best model that maximizes the likelihood. To achieve this, I made 7 models and compared their likelihood to find the best model. Then I tried to find the high probability and low probability 'th' sample using the best model.

3.1 Thresholding:

A threshold can be used to find the correlation for determining the important influencing features. I tried different threshold on the previous task output and observed the independencies of the features.

3.2 Contructing DAG:

Using the information of correlation(independencies), I constructed 7 **DAG** who have directed link between 2 correlated variables and also have minimum edges to cover all 6 features. The models were made manually and used with hit and trail method.

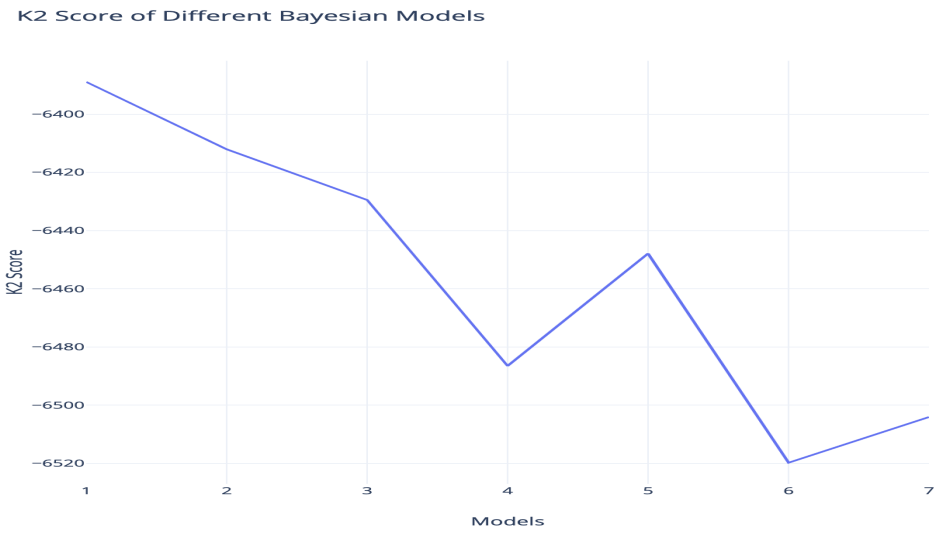
3.3 Ancestral Sampling

As the dataset was not available, we create the dataset from the Model and CPDs using Ancestral sampling.Ancestral sampling is a method to generate dataset when probabilities are for the feature with respect to parent node i.e CPD. In this, the value for the chld node is assigned by sampling the parent using the graph and also does sampling in the top to bottom manner.
To perform Ancestral sampling, I used **forward sampling** defined in pgmpy. In this, we generated a dataset of 1000 for each of the 7 models.

3.4 Likelihood:

As the dataset and model are ready, its time to find out the likelihood of the dataset to the model. I have used **K2Score** to determine the likelihood. K2Score is a scoring function which maps the model to a numerical score, this score is metric to understand how well the model fits the given dataset.// I calculated the K2 score for each of the 7 DAG and then calculated by comparing the scores of the different model to find the best Bayesian model.
The following are the K2 score for 7 Models:

Model	K2 Score
Model1	-6388.92059316
Model2	-6412.07387078
Model3	-6429.4707998
Model4	-6486.50445105
Model5	-6447.90440184
Model6	-6519.75749316
Model7	-6504.08628778



From the table and graph above, we can conclude that model 1 is the best model with the highest K2 Score.

3.5 High and Low Probability 'th':

Finally, from the best Bayesian model, the sampled values were used to find the maximum and minimum frequencies of the unique combination of features values to determine the common sample(high probability) and the rare sample (low probability) of the 'th' was calculated.
High Probability th is:

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

Low probability th is:

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

4 Task 3:

This task required the best best Bayesian network to be converted to the Markov network. In this, I used both the **'th'** and **'AND'** model for inference between Markov Network and Bayesian network in terms of computation time of the models.

4.1 Conversion to Markov Network:

For the conversion of the Bayesian network to Markov network, I used the (BayesianModel.tomarkov) function in pgmpy. This function uses **moralization** for conversion of the network. Moralization is a technique to convert directed graphs into undirected graphs, this converts all the directed edges into undirected edges and also add undirected edges between the parent of the same node.

4.2 Inference using Bayesian Network and Markov Network

In Inference of a network, we answer a probability query for a network given some variable. In this task, I have used **variable elimination** which allows eliminating undesired conditional distribution depending on the dependencies for faster querying of the model. Finally, I queried for computing the conditional probability for variable given parent variable values and measured the time for computing the inference of the model.

Network	Dataset	Time(in sec)
Bayesian Network	'th'	0.00899
Markov Network	'th'	0.01026
Bayesian Network	'AND'	0.015916
Markov Network	'AND'	0.027941

From the above table, we can infer that the Bayesian network is performing better than Markov network for both datasets, this means that Bayesian networks answer the probability query faster for this model. This result is dependent on the network and models, so we cannot generalize that Bayesian network is faster than Markov network always.

5 Task 4:

For this task, I had to create a bayesian model of the "AND" dataset by creating the CPDs from the data and evaluate the goodness score.

5.1 Searching Best DAG:

Searching for best DAG means that exploring the whole search space of possible model and selecting the best model. I have done a heuristic search known as **Hill Climb Search** for getting the best model. Hill climb search is a greedy algorithm which tried to find the **local maximum** by iterating on the edges of the graph to maximize the score and once the local maximum is found then it terminates. For comparison purpose, I also made 4 more models using the dataset, created the CPDs for each model and calculated the K2 score of each model. From all the 5 models, the Hill Climb searched model proved to be the best consisting of all the nodes in the DAG as shown below.

Model	K2 Score
Model1	-9462.70489237
Model2	-9476.24220506
Model3	-9464.57546107
Model4	-9526.33716368
Model5	-9463.48683957

The Edges of the best model are: ('f3', 'f8'), ('f3', 'f9'), ('f3', 'f4'), ('f5', 'f9'), ('f5', 'f3'), ('f9', 'f1'), ('f9', 'f2'), ('f9', 'f4'), ('f9', 'f6'), ('f9', 'f7'), ('f9', 'f8') and its K2 Score using the "AND" dataset is:-9462.70489237.

6 Conclusion:

In this project, I made many Bayesian model from probabilities of 'th' dataset (creating dataset from the probabilities) and "AND" dataset, also compared the K2 score for the model getting the best model for each dataset. I also learned the conversion of directed to undirected graph using Moralization for Bayesian to Markov network, finding the common and rare probabilities from the data.

References

[1] http://pgmpy.org/index.html.

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

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