ADVANCED MACHINE LEARNING - CSE 674

PROJECT 1

Probabilities of Handwriting Formations using PGMs

Sricharan Anand
Department of Computer Science
University at Buffalo,
Buffalo, NY 14214
sanand3@buffalo.edu

OBJECTIVE

The main objective of this project is to build Probabilistic Graphical Models (Bayesian Model, Markov models) to determine probabilities of variables described for the handwritten patterns. The probabilities are used to determine whether the frequency of the handwriting sample, that is to see whether the handwriting is common or rare for an individual. For this project the 'th' and 'and' dataset provided by cedar are considered.

1 INTRODUCTION

1.1 Probabilistic Graphical Models

Formally, a probabilistic graphical model consists of a graph structure. Each node of the graph is associated with a random variable, and the edges in the graph are used to encode relations between the random variables. Depending on whether the graph is directed or undirected, we classify the graphical modes into two types:

- Bayesian Networks.
- Markov Networks.

2 Data

In order to build Probabilistic Graphical Models, we are given the probability distribution for the features of the 'th' handwriting patterns. For the 'th' dataset the marginal distribution and conditional probability distribution for some variables that define the pattern of handwriting sample. The data also includes the CEDAR 'and' dataset which consists of image id and its corresponding features.

3 Implementation

3.1 Task 1

The main aim of Task 1 is to evaluate pairwise correlations and independences that exist in the data. For the given data , we have the Marginal Probabilities Distribution of six different variables which are defined by the pattern of the handwritten sample. This distribution is used to find the independency between different pairs of variables, this can be done by the following equation:

$$\boldsymbol{p}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \boldsymbol{p}(\boldsymbol{x}_i) * \boldsymbol{p}(\boldsymbol{x}_j) \tag{1}$$

Now we also have some of the conditional probability distribution tables that infer probabilities between some of the pairs of discrete variables. The joint probability for the pairs can be calculated using the Conditional Probability Distributions (CPDs) . The joint probability is a key tool for probabilistic inference. Using joint probability, we can learn how events are related probabilistically.

$$\boldsymbol{p}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \boldsymbol{p}(\boldsymbol{x}_i | \boldsymbol{x}_j) * \boldsymbol{p}(\boldsymbol{x}_j) \tag{2}$$

For multi-categorical variables, which is in the given data we calculate the correlation between different variables :

- Calculate cross entropy between the variables, that is between P(x,y) and P(x)P(y)

  This can be done by taking:

```
x2|x1 : 0.15977
x4|x1 : 0.11943000000000004
x6|x1 : 0.16015500000000005
x3|x2 : 0.21852500000000002
x5|x2 : 0.12926000000000004
x2|x3 : 0.21875800000000006
x5|x3 : 0.11551999999999997
x6|x3 : 0.09564000000000002
x1|x4 : 0.11957000000000002
x2|x4 : 0.11569999999999997
x6|x4 : 0.14346999999999996
x2|x5 : 0.13126499999999997
x3|x5 : 0.11596500000000005
x1|x6 : 0.16036999999999996
x2|x6 : 0.17531500000000003
x3|x6 : 0.13116000000000005
x4|x6 : 0.14307000000000003
```

Fig 1: Correlation between variables

$$\sum \quad \boldsymbol{abs}((\boldsymbol{P}(\boldsymbol{x}_i | \boldsymbol{x}_j) - \boldsymbol{P}(\boldsymbol{x}_i))) \tag{3}$$

From the above results we see that we don't need to discard any pairs as they have a low and approximately close values, so we can infer that the variables are independent.

3.2 Task 2

In task 2, we construct a bayesian network with the edges that give maximum likelihood. For this we first draw links between different correlated pairs of variables. The bayesian graphical model helps us represent complex networks of interrelated and independent events efficiently. Bayesian models use directed graphs and are used when there are casual relationships between the random variables. Here we construct bayesian graphs based on the pairwise correlation between the variables. There must not be any loops while constructing the networks.

Bayesian network is a directed acyclic graph in which each node corresponds to a conditional dependency, and each node corresponds to a unique random variable. For example in our data, if there is edge between x_1 and x_2 such that there is a direction from x_1 to x_2, then we get the factor $P(x_2|x_1)$. Example of a graph of BN for Fig 2 factorizes the joint distribution as:

$$p(x_1, x_2, x_3, x_4, x_5, x_6) = p(x_1)p(x_6|x_1)p(x_4|x_1, x_6)p(x_2|x_6)p(x_3, x_2)p(x_5|x_2)$$

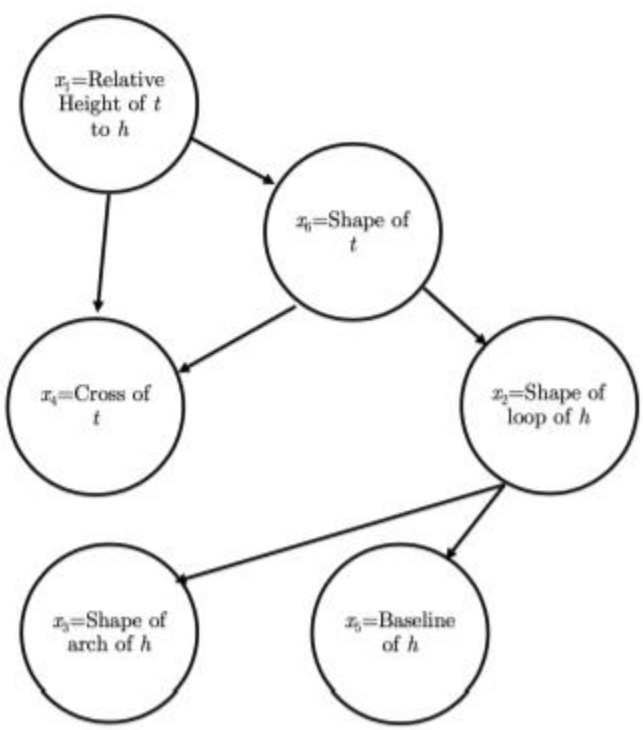

Fig 2.

The next step is to do ancestral sampling, where we perform sampling for size of 1000. The distribution is specified by

$$\boldsymbol{p(z) = \prod_{i=1}^{M} p(z_i | pa_i)}$$

where z_i are set of variables associated with node *i* and pa_i are the set of variables associated with the node parents of *i*. After one pass through the BN, we get a sample.

Finally, we can now calculate K2 score for all the networks. The best bayesian network has the highest K2 score. For my sample network, the best bayesian network was:

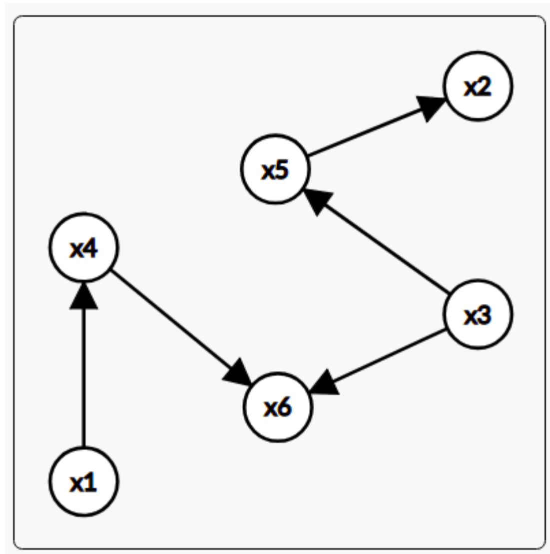

Fig 3.

```
-6489.736764997548
-6466.482231659014
-6406.280284514297
-6474.261280170384
-6504.329885189505
K2 score for Best Bayesian newtwork: -6406.280284514297
```

Fig 4.

3.3 Task 3

The aim for task 3 is to convert the best Bayesian model to markov model. A PGM is called Bayesian network when the graph is directed whereas in Markov network/ Markov random field the graph is undirected. In this project as the graph is directed, we do not see much of a difference.The markov model generated will be the moral graph of the bayesian model.

```
Accuracy for BayesianModel:  78.10000000000001
CPU times: user 6.77 s, sys: 120 ms, total: 6.89 s
Wall time: 6.76 s
```

Fig 5.

```
Accuracy for Markov 78.10000000000001
CPU times: user 5.48 s, sys: 117 ms, total: 5.6 s
Wall time: 5.46 s
```

Fig 6.

By the method of inference, that is Belief Propagation we can find the accuracy of the model. From fig 5 and fig 6 we can see that by inference (MAP_query) the accuracy is the same, but the difference is in the time it takes to traverse the graph and generate output. Markov model takes lesser time to run in comparison to Bayesian Network.

3.4 Task 4

In task 4, the goal is to construct the best Bayesian network for the AND dataset and evaluate the goodness score. In this task we use HillClimbSearch algorithm. This algorithm finds the best Bayesian network among different combinations of network. In numerical analysis, hill climbing is a mathematical optimization technique which is an iterative algorithm that starts with an arbitrary solution to a problem then attempts to find a better model by making incremental changes to the solution.

For goodness score, we can use K2 score to find best most. By the HillClimbSearch we get the best model with the best K2 score. In order to compare, new networks were created for the 'AND' dataset.

```
K2 Score for Best model('AND' Dataset): -9462.704892371386
K2 Score for Model 1: -9685.415038946254
K2 Score for Model 2: -9903.963188099286
K2 Score for Model 3: -9932.012183623841
K2 Score for Model 4: -9863.836347673107
```

Fig 7.

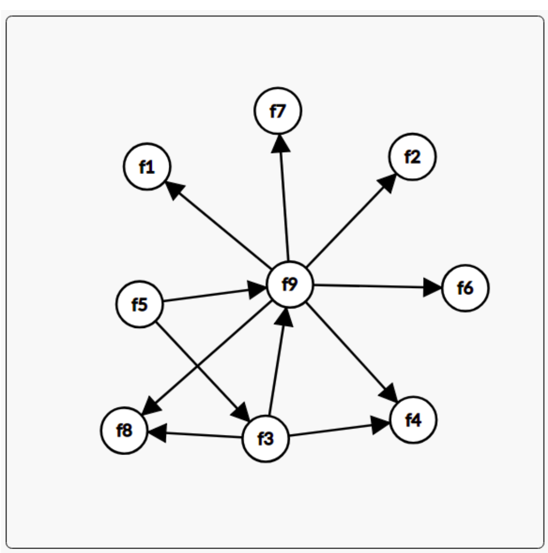

Fig 8.

CONCLUSION

In conclusion, probabilistic graphical model are less sensitive to noise content. The difference between other Machine Learning algorithm and PGM's are that, PGM's provide a more structured learning and give a intuitive relationship between variables. When there is a dependence between variables, graphical models can help reduce the computation required to infer something. In this project we built bayesian models for the 'th' and 'AND' dataset which helped us compute faster as the graphical models are faster when there are dependencies given. Hence it reduces the presence of uncertainty.

References

1. http://piazza.com/redirect/s3?bucket=uploads&prefix=attach%2Fjq8nawb0m0v7hy%2Fisamd3soc56z%2Fjs534pn8hni3%2FPGMPY_tutorial.zip
2. http://pgmpy.org/