Advanced Machine Learning Project 1 :

Determining Probabilities of Handwriting Formations using PGMs

Rohan Hemanshu Sheth

UID: 50291746

Objective:

The objective of this project is to develop probabilistic graphical models (PGMs) to determine probabilities of observations which are described by several variables. The project only concerns with the letter-pair "th" for determining how frequent is the occurrence and the probability of a given pair to be belonging to a certain individual. A characterization of the structure of th as given by document examiners (human experts) is provided. In this characterization there are six random variables x1-x6. Variable xi can take one of a set of discrete values, denoted as x j i. The dataset consists of 6 of the possible conditional distributions and one marginal distribution of the variables. We need to provide inferences based on the generated PGMs.

Task 1:

The following are the calculated existing pairwise correlations in the data. Correlation between two variables can be calculated as Correlation(xj, xi) = $\sum$abs((P(xj|xi)P(xi) – P(xi)P(xj)). We assume independence for pairs of variables for which probabilistic relations are not provided in the data tables. We can determine if xi and xj are independent by checking if p(xi , xj) = p(xi)p(xj), where P(xi , xj) = P(xi |xj)P(xj) from the tables. Since some of the conditional probability distribution tables turned out to be invalid, some of the Table data has been changed manually to satisfy the sum of probabilities to be 1.

Table 3 :

```
Correlation(x2,x1):  0.15977
Correlation(x4,x1):  0.11943
Correlation(x6,x1):  0.16015500000000008
```

Table 4:

```
Correlation(x3,x2):  0.21852500000000002
```

```
Correlation(x5,x2):  0.12926000000000004



Table 5:

Correlation(x2,x3):  0.218758
Correlation(x5,x3):  0.11551999999999996
Correlation(x6,x3):  0.09564

Table 6:

Correlation(x1,x4):  0.11956999999999998
Correlation(x2,x4):  0.11569999999999998
Correlation(x6,x4):  0.14346999999999993

Table 7:

Correlation(x2,x5):  0.131265
Originally, this value was 0.856145
Correlation(x6,x5):  0.115965

Table 8:

Correlation(x1,x6):  0.15259999999999999
Correlation(x2,x6):  0.17531500000000005
Correlation(x3,x6):  0.13903000000000001
Correlation(x4,x6):  0.14307000000000003
```

Task 2:

After constructing the required conditional probability distribution tables of conditional probabilities from tables 3 to 8 as well as marginal probabilities from table 2, different Bayesian networks are constructed based on the most correlated pairs obtained from Task 1. We need to make the model such that the likelihood is maximized with the networking containing as few edges as possible. This is done using ancestral sampling where the network assigns a likelihood score to atleast a 1000 samples. The Bayesian networks are generated by forming links between the most correlated variables and a K2 score is assigned to each. Here, the dataset changes for each model because the directed links mean different marginal and conditional distributions for that model.

Here are 5 of the models that were generated:

Model 1:

network.add_edges_from([('x1','x2'),('x1','x4'),('x1','x6'),('x2','x5'),('x2','x3')])

K2 score : `-6367.944234899049`

Model 2:

network1.add_edges_from([('x2','x3'),('x2','x5'),('x3','x6'),('x6','x1'),('x6','x4')])

K2 score : -6338.962289899204

Model 3:

network2.add_edges_from([('x1','x4'),('x1','x2'),('x2','x3'),('x4','x6'),('x3','x5')])

K2 score : -6451.003840774321

Model 4:

network3.add_edges_from([('x3','x2'),('x2','x5'),('x3','x6'),('x6','x1'),('x6','x4')])

K2 score : -6454.072380638447

Model 5:

network4.add_edges_from([('x6','x2'),('x2','x5'),('x5','x3'),('x6','x4'),('x4','x1')])

K2 score : -6383.26456557056

Now all the samples are combined and K2 scores are assigned to the models based on the combi ned data, they are:

```
-31922.465946215812
-31906.205802519173
-31942.598709225465
-31906.077405075514
-32141.20688823206
```

Best Bayesian Network is Model 4 since its K2 score is the highest while the worst Bayesian Net work is Model 5 since its K2 score is the lowest among the assigned values.

Task 3:

We convert the best Bayesian model obtained in Task 2, Model 4, into a Markov Model using moralization and belief propagation and obtain a likelihood score for both of them for compariso n.

Computation Time for Best Bayesian Network converted to Markov Network : 0.028000831604003906 secs

Computation Time for Best Bayesian Network: 0.012248516082763672 secs.
Markov Network takes more time than the corresponding Bayesian Network. It is slower. Both models have equal performance in terms of accuracy.

Task 4:

We use “AND-Features” dataset to construct several Bayesian Networks and assign a likelihood score to determine the best model. In order to determine the best model, we use Hill Climb Search to estimate the best model based on locally maximum K2 score of the data.

Here are the models:

model = BayesianModel([('f3', 'f4'), ('f3', 'f9'), ('f3', 'f8'), ('f5', 'f9'), ('f5', 'f3'), ('f9', 'f8'), ('f9', 'f7'), ('f9', 'f1'), ('f9', 'f6'), ('f9', 'f2'), ('f9', 'f4')])

k2 score = -9462.704892371386

model1 = BayesianModel([('f1', 'f2'), ('f1', 'f3'), ('f2', 'f4'), ('f4', 'f5'), ('f4', 'f8'), ('f3', 'f6'), ('f6', 'f9'), ('f9', 'f7')])

k2 score = -9812.171135684388

model2 = BayesianModel([('f1', 'f6'), ('f1', 'f8'), ('f6', 'f2'), ('f6', 'f3'), ('f8', 'f5'), ('f8', 'f4'), ('f2', 'f7'), ('f5', 'f9')])

k2 score = -9832.172843143524

model3 = BayesianModel([('f1', 'f9'), ('f1', 'f4'), ('f9', 'f7'), ('f4', 'f8'), ('f8', 'f6'), ('f6', 'f2'), ('f6', 'f5'), ('f5', 'f3')])

k2 score = -9757.454830245364

Based on the k2 scores, the model based on Hill Climb Search is the best model. Estimated conditional probability distribution of features like 'f9' based on the model and data-set are also generated.