Assignment 1: Bayesian Network

Kazi Injamamul Haque

November 2018

1 Introduction

This assignment compares different learning methods of Bayesian Network on the given dataset, ’leukemia.dat’. The three learning methods for comparisons are- NPC, Greedy Score-And-Search and Naive Bayes Structure. For learning and analyzing the results, Hugin Lite 8.3 was used and a simple python script was used to split the dataset into train and test datasets. The methodology of splitting the data and learning the models using the software are described shortly in the next section. The obtained results are explained in the ”Results and Discussion” section. Finally the summary of the comparison is shown in the last section ”Conclusion”.

2 Methodology

The given data was divided into train and test datasets. The training dataset, ’leukemia_train.dat’ consists of 80% data from the original dataset and the test dataset, ’leukemia_test.dat’ consists the rest. The simple python script, ’split.py’ that used for splitting the data is shown below where pandas was used to manipulate data and sklearn was used for splitting.

```
// split.py

import pandas as pd
from sklearn.model_selection import train_test_split


data = pd.read_csv('leukemia.dat')
train, test = train_test_split(data, test_size=0.2)
train.count()
test.count()
train.to_csv('leukemia_train.dat', index = False)
test.to_csv('leukemia_test.dat', index = False)
```

For learning and analyzing different models, the same train and test datasets were used for simplicity and comparison. In order to learn the models, similar method was used within the given software framework. Learning Wizard in hugin was used to import the training set, set up the network and learn the models given some additional parameters. For NPC(Necessary Path Condition), all the default values were used including the default threshold and it produced the network shown in Figure 1. For Greedy Search-And-Score learning, similar process was followed, maximum parents for nodes set to 7 and default settings was used as well. Figure 2 depicts the learned network structure from greedy search-and-score learning. Naive Bayes structure assumes that the label(in this case AML) depends on all the predictors separately and there is no dependencies on each other for the predictors. Therefore, for learning Naive Bayes network, the path constraints were defined by the user and then EM-Learning wizard was used to train the model using the same train dataset as before. The network is shown in figure 3 for Naive Bayes Structure.

3 Results And Discussion

3.1 NPC

3.1.1 Learning

NPC model was learned with 57 cases (80% of the original data). After three iteration, the log-liklihood converges to -150.432 and yields AIC and BIC score of -167.432 and -184.797 respectively. Figure 1 shows the final outcome of the learning.

Learning with	57 cases
Log-likihood	-150.432
AIC	-167.432
BIC	-184.797

3.1.2 Analysis

The confusion matrix obtained from the analysis wizard of the software depicts the accuracy of the learned model. The NPC learning gives 20% error in classifying the labels from the test dataset which consists of 20% or the original data. For fifteen test cases, NPC model correctly classifies 12 cases and gave 3 true-negative cases. The average Euclidia distance ins 0.22652 and the average Kullbach-Leibler divergence is 0.32412.

Actual	YES	NO	Predicted
-	4	3	YES
-	0	8	NO

Table 1: NPC Confusion Matrix

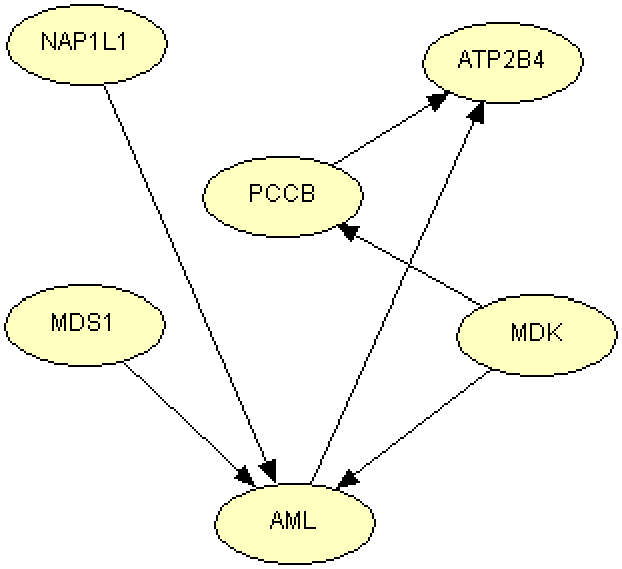

Figure 1: Learned NPC Network

Number of cases	15
Using cutoff threshold	Max. belief
Error rate	20.00
Avg. Euclidian distance	0.22652
Avg. Kulbach-Leibler divergence	0.32412

3.2 Greedy Search-And-Score

3.2.1 Learning

Greedy Search-And-Score learning also had the same training and testing datasets. With 57 cases used for training, the Log-liklihood was -146.33 after 3 iteration. The AIC and BIC scores were -159.33 and 172.61 respectively. The network learned from using this learning algorithm is shown in Figure 2.

Learning with	57 cases
Log-likihood	-146.33
AIC	-159.33
BIC	-172.61

3.2.2 Analysis

’leukemia_test.dat’ was used to test the model that consists of 15 cases with known labels. The confusion matrix shown in table 2 shows the accuracy of the

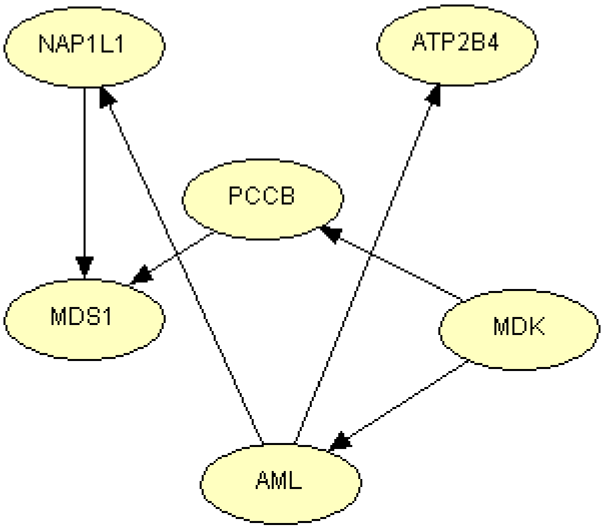

Figure 2: Learned Greedy Search-And-Score Network

learned model. Eleven cases were predicted correctly out of fifteen. Four cases were classified as positive those in reality are negative(true-negative) and none as false-positive. The error rate is 26.67%, average Euclidian distance is 0.28697 and the average Kullbach-Leibler divergence is 0.40629.

Actual	YES	NO	Predicted
-	4	4	YES
-	0	7	NO

Table 2: Greedy Search-And-Score Confusion Matrix

Number of cases	15
Using cutoff threshold	Max. belief
Error rate	26.67
Avg. Euclidian distance	0.28697
Avg. Kulbach-Leibler divergence	0.40629

3.3 Naive Bayes Structure

3.3.1 Learning

EM-Learning was used as opposed to structured learning for this case. The user defined the dependencies of the output label on all the predictors assuming there is no dependency on each other for the predictors. The network is shown in Figure 3. The Naive Bayes Structure was learned with 57 cases with initial experience tables set to 1/Pa(x) for all the nodes, where x is the node itself. The Log-liklihood yields -154.275 after 3 iterations. The AIC and BIC scores recorded as -165.275 and -176.512 respectively.

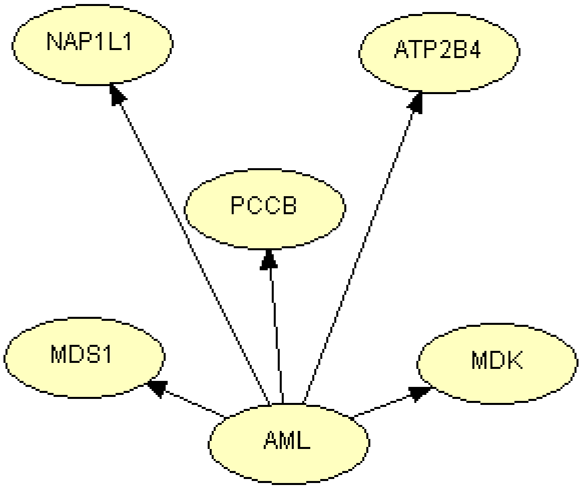

Figure 3: Naive Bayes Structure

Learning with	57 cases
Log-likihood	-154.275
AIC	-165.275
BIC	-176.512

3.3.2 Analysis

The analysis of the trained model was done with the test data consisting of 15 cases. The confusion matrix shown in Table 3 depicts that the naive bayes method of learning correctly predicted 13 out of 15 cases in the test data. Only 2 of the cases were classified as positive those were actually negative. Therefore, the error rate is 13.33%. The average Euclician distance is 0.21064 and the average Kullbach-Leibler divergence is 0.32169.

Number of cases	15
Using cutoff threshold	Max. belief
Error rate	13.33
Avg. Euclidian distance	0.21064
Avg. Kulbach-Leibler divergence	0.32169

Actual	YES	NO	Predicted
-	4	2	YES
-	0	9	NO

Table 3: Naive Bayes Structure Confusion Matrix

4 Conclusion

Table 4 compares the results of the three learning models and shows the error rate for the same test dataset. Greedy Search-And-Score method shows that the Log-liklihood, AIC and BIC scores are better that the other two as these scores are closer to 0 than those of NPC and NB. On the other hand, Naive Bayes Structure shows the least error while classifying test data. The dataset used for this assignment was very small and might be biased as well. If the dataset contained a lot more data than what was given, the models are most likely to change along with the error rates. As long as the error rate is concerned, using a Naive Bayes Structure learning is more efficient given the data that was provided.

Models	Log-liklihood	AIC	BIC	Error Rate
NPC	-150.432	-167.432	-184.797	20.00
Greedy	-146.33	-159.33	-172.61	26.67
NB	-154.275	-165.275	-176.512	13.33

Table 4: Comparison of the models

5 Online References

1. **http://download.hugin.com/webdocs/manuals/Htmlhelp/descr_NPC_algorithm_pane.html**
2. **http://download.hugin.com/webdocs/manuals/Htmlhelp/descr_greedy_pane.html**
3. **https://en.wikipedia.org/wiki/Naive_Bayes_classifier**
4. **https://disi.unitn.it/ passerini/teaching/2018-2019/MachineLearning/**