UNIVERSITÀ DEGLI STUDI DI TRENTO

Department of Information Engineering and Computer Science

Master Degree in
Computer Science

Bayesian Network Report
Hugin

Professor
Andrea Passerini

Assistant
Luca Erculani

Student: Enrico Saccon
ID: 207531

Anno accademico 2019/2020

Introduction

The goal of the project is to learn a Bayesian Network structure from a dataset using different algorithms and then test the obtained network.
The dataset is composed of a total of 72 examples. Of the total dataset, 80%, i.e., 58 examples, will be used for training and the remaining, i.e., 14 examples, will be used for testing.
Each line in the dataset is made of 6 entries of which 5 represent the state of 5 different genes and one entry says which is the type of leukemia (AML/ALL).
The algorithms that are going to be tested are:

- NPC;
- Greedy search-and-score;
- Fixed Naive Bayes structure.

Training

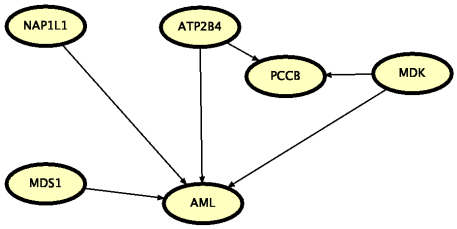
It's possible to create the network by selecting the correct learning algorithm when using the provided wizard.
Once the training set is loaded, it's possible to give constraints to the structure, for example the fact that some links do not exist, or if we know that some other exist.
Then after choosing the wanted structure learning algorithm we are given the preview of the Bayesian Network and the possibility to add some experience tables. This is helpful for testing afterwards. Each time the experience was initialized to a default value of 0.1. This is done to avoid the possibility of having some probabilities estimated to 0 with the maximization of the likelihood.
At last one selects the number of iterations, e.g., 5, of the expectation-maximization algorithm which is guaranteed to converge to a local minimum.

Testing

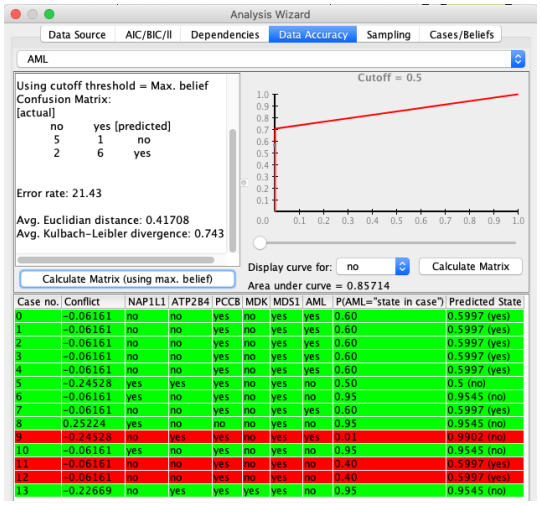
The testing phase is done by using the provided analysis wizard.
Once invoked, it allows one to choose the test set and test whether a combination of genes would be correctly predicted.
The cutoff threshold to decide about the correctness is set using the maximum belief option.
The wizard gives also the confusion matrix of the test which allows to compute values such as

- Accuracy
$$Acc = \frac{TP + TN}{TP + TN + FP + FN}$$
- Precision
$$Prec = \frac{TP}{TP + FP}$$
- Recall
$$Rec = \frac{TP}{TP + FN}$$
- F-measure
$$F_1 = \frac{2(Prec \times Rec)}{Prec + Rec}$$

NPC



As for the testing, an error rate of 21.43% was obtained while the confusion matrix is shown in the following picture.



True \ Pred	Yes	No
Yes	6	2
No	1	5

From this it's possible to compute scoring values:

$$Acc = \frac{6+5}{6+5+1+2} = \frac{11}{14} = 0.78$$

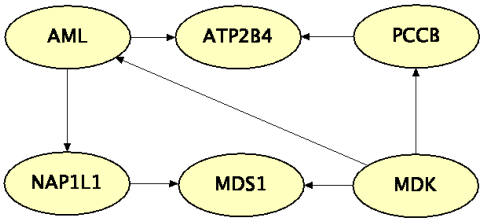
The accuracy is actually 1 minus the error rate.

$$Prec = \frac{6}{6+1} = 0.85$$

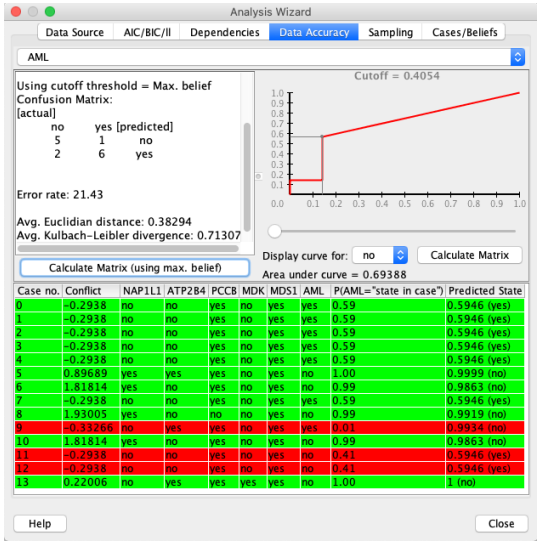
$$Rec = \frac{6}{6+2} = 0.75$$

$$F_1 = \frac{2(0.85+0.75)}{0.85+0.75} = 0.8$$

Greedy Search-and-Score



As for the testing, an error rate of 21.43% was obtained while the confusion matrix is shown in the following picture.



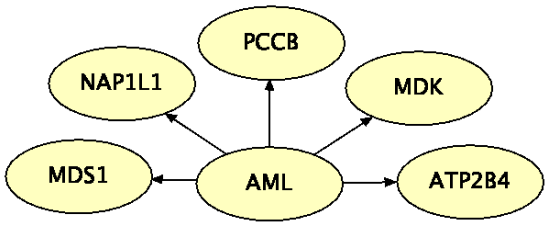
True \ Pred	Yes	No
Yes	6	2
No	1	5

The confusion matrix is the same as before, even though the structure is quite different from before. It's also possible to see that the curve is quite different and the maximum belief is lower, set to 0.4 instead of 0.5. This could be due to the fact that the MDK gene controls greatly the result of AML: indeed, if MDK is set as evidence in Hugin, the probability of obtaining `no` in AML is 99.65%

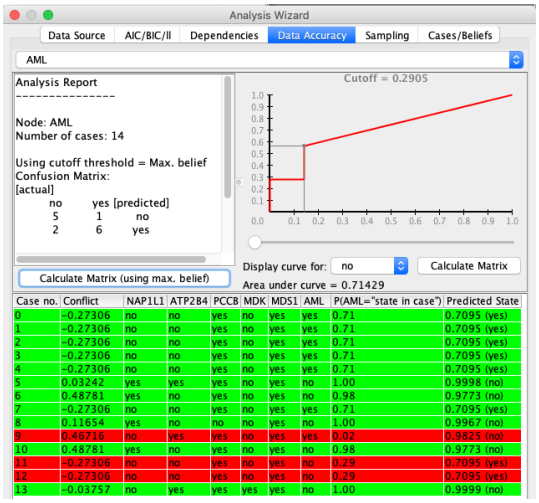
$$Acc = \frac{6+5}{6+5+1+2} = \frac{11}{14} = 0.78 \qquad Prec = \frac{6}{6+1} = 0.85$$

$$Rec = \frac{6}{6+2} = 0.75 \qquad F_1 = \frac{2(0.85+0.75)}{0.85+0.75} = 0.8$$

Fixed Naive Bayes



As for the testing, an error rate of 21.43% was obtained while the confusion matrix is shown in the following picture.



True \ Pred	Yes	No
Yes	6	2
No	1	5

The confusion matrix is the same as the two previous examples, even though the structure is different again.

$$Acc = \frac{6+5}{6+5+1+2} = \frac{11}{14} = 0.78 \qquad Prec = \frac{6}{6+1} = 0.85$$

$$Rec = \frac{6}{6+2} = 0.75 \qquad F_1 = \frac{2(0.85+0.75)}{0.85+0.75} = 0.8$$

Conclusions

All the three learning algorithms return similar results. This could be due to a gene controlling the AML result more strictly or to the fact that actually the dataset is too small to show differences in the learning algorithm.