BRL Assignment 2

Arne Lescrauwaet (852617312)

June 2024

1 Introduction

In this report, we take a look at two experiments. The first experiment compares two algorithms and their ability to learn a structure from a dataset. The second experiment focuses on the evaluation of a learned classifier and possible ways of improving the structure when learning from data.

1.1 Effect of Sample Size on Learning Structure

In this segment of the assignment, we explore the influence of sample size on the ability to discern the structure of a Bayesian Network. We compare the performance outcomes of a constraint-based learning algorithm (greedy hill climb) and a search-and-score algorithm (MIIC) across varying sample sizes: 1, 10, 100, 500, 1000, 1500, 2000, 2500, 3000, and the full dataset consisting of 3654 samples. These resultant networks are subsequently compared to the manually constructed rain prediction network from Assignment 1.

1.2 Learning a Classifier from Data

In this section of the assignment, we examine the capacity of the hill climb algorithm to construct a network based on a given dataset. The resultant learned network is then compared to a network constructed by an expert, with both networks being evaluated using the receiver operating characteristic (ROC) curve. Finally, we explore the potential of bringing the learned network closer to the expert network by incorporating constraints.

2 Methods

2.1 MIIC and GHC on different sample sizes

Before employing the MIIC and GHC algorithms to learn a network from the data, we must first establish a template. This template includes all the variables from our target network, as illustrated in Figure 1. Subsequently, we need to define the sample sizes to be used in our experiment. We selected the following sample sizes: 1, 10, 100, 500, 1000, 1500, 2000, 2500, 3000, and the full dataset (3654). This range was chosen to effectively capture the potential impact of varying sample sizes on the learning algorithms, from minimal data to the complete dataset.

Figure 1: Rain prediction template

The structure can be learned using the algorithms in the following manner: First, define a learner with the command ghc/miic = gum.BNLearner(data, template). Then, select the algorithm by using the corresponding functions: ghc.useGreedyHillClimbing() or miic.useMIIC(). Finally, learn the structure with net = ghc/miic.learnBN().

After iterating over the sample sizes and learning the structures using our two algorithms, we need a method to compare them. For this purpose, we selected the structural Hamming distance (SHD). The SHD is a metric used to compare the structures of two Bayesian networks. It measures the minimum number of operations (additions, deletions, or reversals of edges) required to transform one network structure into another. The SHD can be easily computed using the gum.lib.bnvsbn.GraphicalBNComparator(net1, net2) function and accessed by using the .hamming() function on the result of the previous function. These SHDs were then analyzed to assess the impact of different sample sizes on the ability of the MIIC and GHC algorithms to learn the network structure.

2.2 GHC as a classifier

Similarly to the previous section, before we can learn a structure from our data, we need to establish a template. This time, we loaded the target network using a ".net" file, necessitating the extraction of variables for our template. We achieved this by iterating over the names in the network and utilizing the net.variableFromName(name) function.

The process of learning the structure using the GHC algorithm was identical to the previous section, but this time we utilized the full dataset instead of varying sample sizes. After learning the structure, we compared it to the expert network using the SHD and visually with the bnvsbn.graphDiff(net1, target) function, which highlights missing, reversed, or additional connections. The learned network was also evaluated and compared to the expert network in terms of classification performance using the ROC curve metric via the bn2roc.showROC(net, data, variable, value) function. In the final part of this experiment, we attempted to enhance the structure of the learned network by manually introducing constraints with the ghc.addMandatoryArc(variable1, variable2) function and assessed whether these modifications improved the structure and the ROC curve. The complete implementation can be found on Github.

3 Results

3.1 Results of Sample Sizes on Learning Structures

My hypothesis prior to starting the experiment was that similar to machine and deep learning, larger sample sizes would generally yield better results than smaller ones. After learning and evaluating the structure from various sample sizes, we observed that the SHD tends to decrease for both algorithms as the sample size increases. Notably, we observed a peak SHD of 8 for the MIIC algorithm at a sample size of 3000, and another peak SHD of 8 for the GHC algorithm at a sample size of 3500. The causes of these spikes are not immediately clear and may be attributed to the sampling process, sampling more of one value than others causing the algorithm to learn wrong interactions.

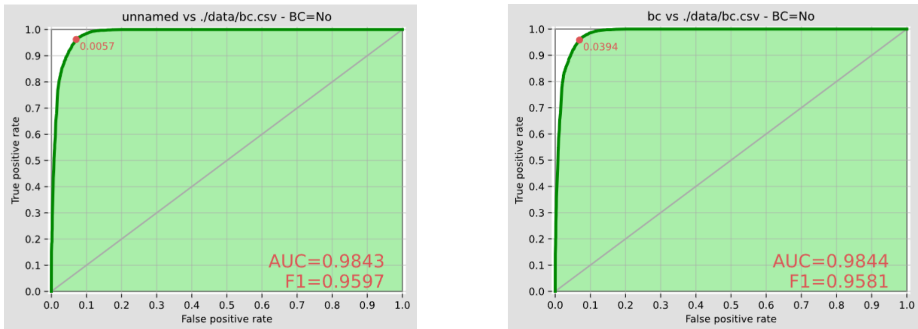
The SHD appears to stabilize at a value of 5 for both algorithms around 1500 - 2000 samples, with little improvement for larger sample sizes, as depicted in Figure 3. There does not seem to be a significant difference between the MIIC and GHC algorithms, as both reach a minimum SHD of 5 at approximately the same sample size. However, one could argue that the MIIC algorithm has a slight advantage, as it exhibits a lower average SHD of 6.1 compared to 6.8 for the GHC algorithm.

3.2 Results of Learning a Classifier

The initially learned network differed from the expert network by an SHD of 9. It had one missing connection (indicated by a red dotted arrow) and four reversed connections (indicated by purple arrows), as illustrated in Figure 4.

Although the learned model differs slightly from the expert model, their performance on the ROC curve metric for binary classification of the breast cancer (BC) variable is practically identical. The learned model achieved an AUC of 0.9843 and an F1 score of 0.9597, compared to 0.9844 and 0.9581 for the expert model, as shown in Figure 2a.

An attempt to improve the structure of the learned network was made by introducing constraints. In the expert network, both "Age" and "Location" influenced "BC". This relationship was not captured in the learned network, as shown in Figure 6. To address this, the constraints "Age" must influence "BC" and "Location" must influence "BC" were introduced to force the model to learn these relations. This adjustment reduced the SHD of the learned model compared to the expert model from 9 to 2. Additionally, the ROC curve metric showed a slight improvement, with the AUC increasing from 0.9843 to 0.9844 and the F1 score improving from 0.9597 to 0.9584. This resulted in the learned model achieving the same AUC and an even better F1 score than the expert model.



(a) ROC curve of learned model

(b) ROC curve of expert model

Figure 2: Learned model ROC curve vs Expert model ROC curve

4 Conclusions

It appears that a larger sample size indeed enhances the ability of the MIIC and GHC algorithms to learn a structure. It would be intriguing to investigate whether the algorithms would continue to improve their SHD with additional data or if the SHD of 5 represents the minimum achievable value for these learning algorithms on this dataset.

It is interesting to note that the paper ”Learning Bayesian Network for Rainfall Prediction Modeling in Urban Area using Remote Sensing Satellite Data” by S. Putri et al. [1] also employs variants of the Hill Climb algorithm (Regular Hill Climbing and Maximum-Minimum Hill Climbing) on a similar dataset but produces a markedly different network although their network is similar to the one used in Assignment 1.

Despite differing from the expert network by an SHD of 9, the learned network exhibited an almost identical ROC curve and F1 score. This demonstrates that using learning algorithms is a valuable method for constructing networks, especially when there is a lack of domain knowledge. Nevertheless, domain knowledge can be easily integrated into algorithmic learning by introducing constraints.

References

[1] Salwa Rizqina Putri and Arie Wahyu Wijayanto. “Learning Bayesian Network for Rainfall Prediction Modeling in Urban Area using Remote Sensing Satellite Data (Case Study: Jakarta, Indonesia)”. In: *Proceedings of The*

International Conference on Data Science and Official Statistics 2021.1 (Jan. 2022), pp. 77–90. DOI: 10.34123/icdsos.v2021i1.37. URL: https://proceedings.stis.ac.id/icdsos/article/view/37.

5 Appendix

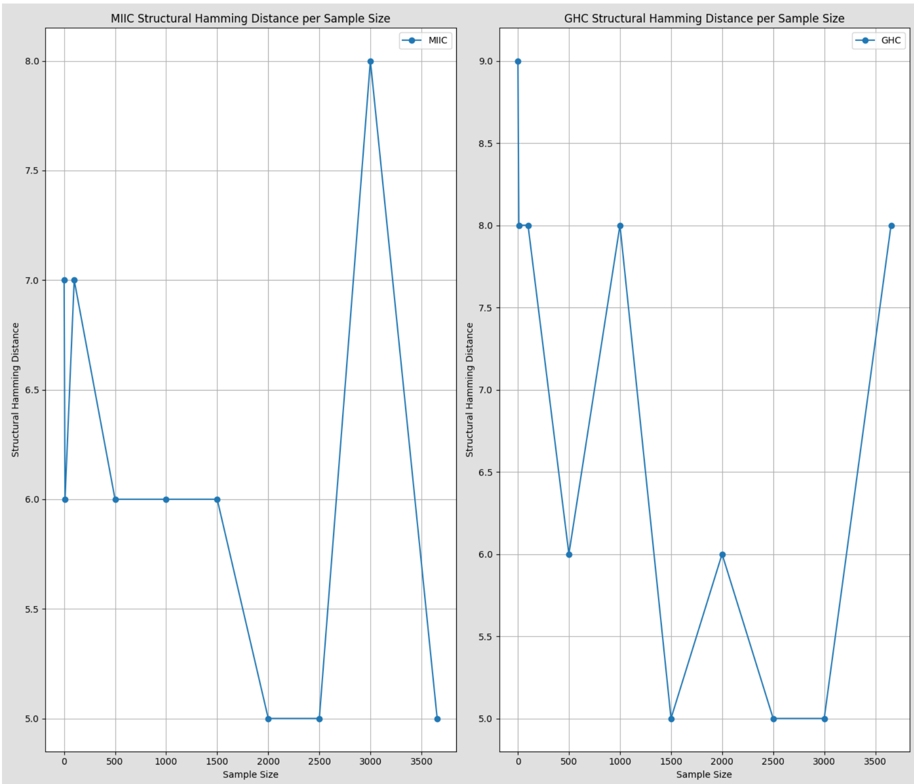


Figure 3: SHD for the MIIC and GHC algorithm per sample size

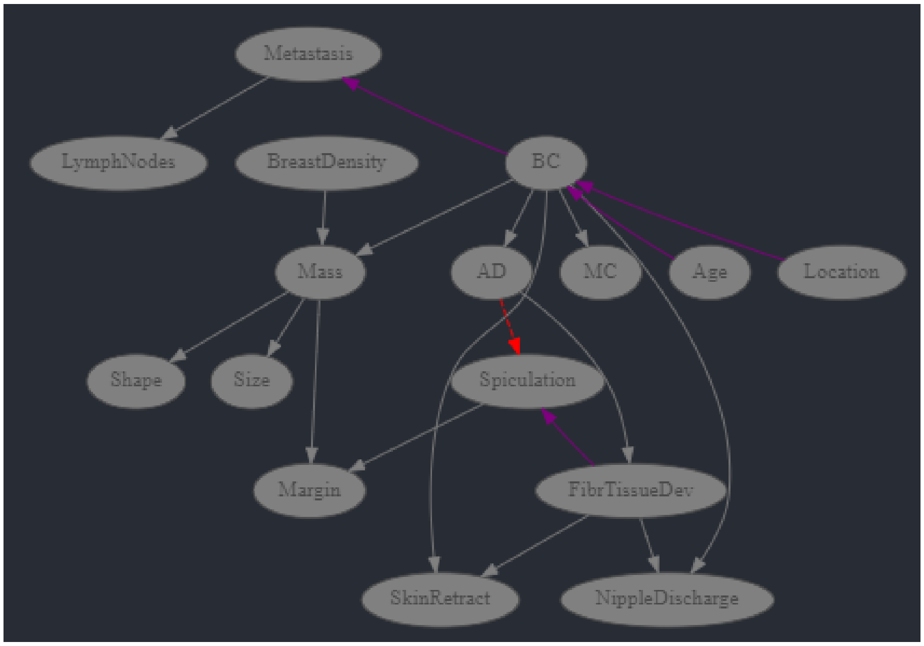
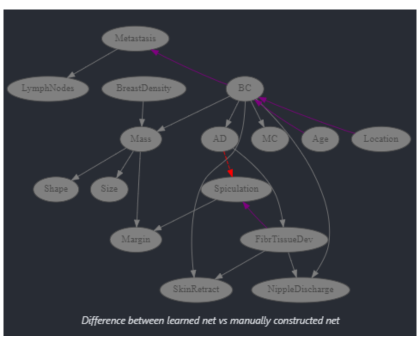
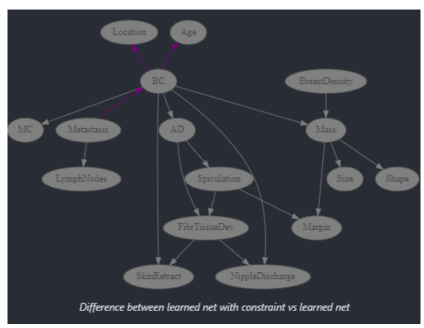


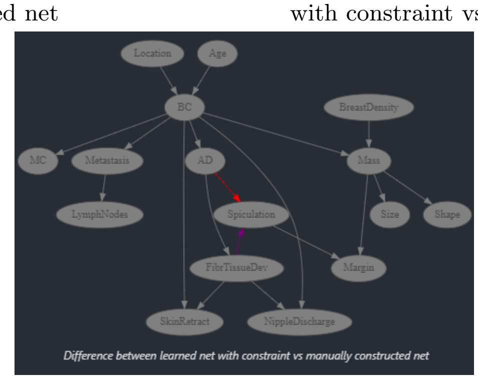
Figure 4: Difference between learned network vs manually constructed network



(a) Difference between learned net vs manually constructed net



(b) Difference between learned net with constraint vs learned net



(c) Difference between learned net with constraint vs manually constructed net

Figure 5: Learned vs constraint vs expert

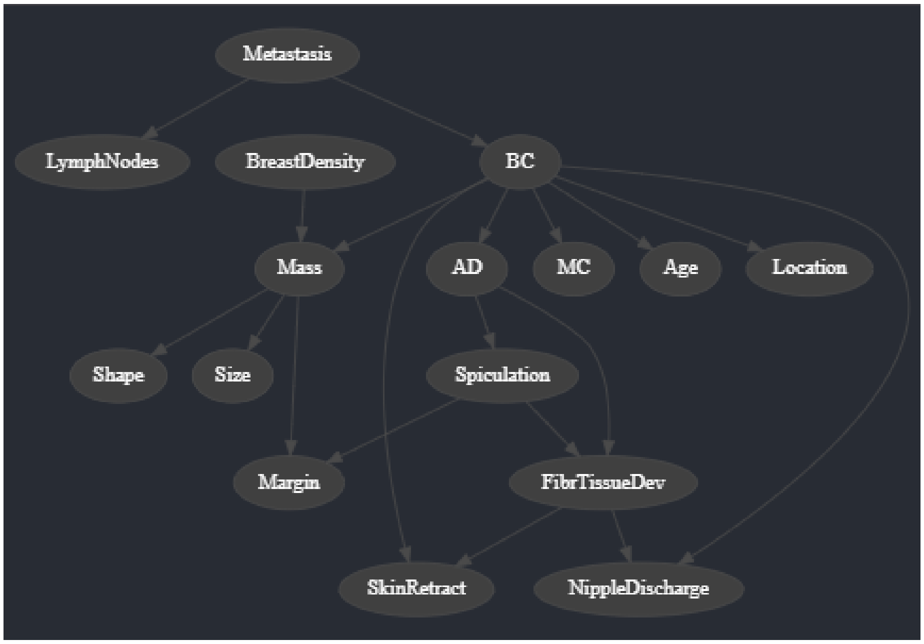


Figure 6: Learned network by the GHC algorithm on the BC dataset