Assignment 2 report

Bayesian Networks and Causal Inference

Floris Rossel (s1010794), Andro Radičević (s1102974), Mario Tsatsev(s1028415)

January 26th, 2023

1 Introduction

Automated inference of a Bayesian network structure from data is a common challenge in the field of Bayesian networks. Although in many cases a network can be constructed manually using domain knowledge and tested against data, this is not always possible or practical. Therefore, many algorithms have been devised in order to learn the structure of networks directly from data. In this assignment, we will evaluate two types of structure learning algorithms: constraint-based (represented by the PC algorithm) and score-based (represented by the hill climbing algorithm). We aim to explore how variations in the hyperparameters of these algorithms can impact the accuracy of the inferred network structure and identify optimal parameter settings for these algorithms.

For demonstration of the structure learning algorithms we used the "Higher Education Students Performance Evaluation Dataset"[6]. It contains 145 samples of 33 attributes each concerning the study habits and background information of students, as well as their grades. All attributes are categorical.

2 Methods

We compare two algorithms, one constraint-based, and one score-based algorithm. For the constraint based algorithm we use the PC algorithm. For the score-based approach, we use a greedy hill-climbing algorithm using two different scoring functions; AIC and BIC.

Before the data could be used in algorithms, it had to be preprocessed. First, we removed most of the data concerning the background information of students, leaving only the basic background information of age and sex, in order to focus on the impact of various study habits. We also changed some of the variables to ordinal where it made sense, and others to binary. The table describing the final dataset in detail is presented in the appendix. 2

2.1 PC-algorithm

The PC (Parents-Children) Algorithm is a constraint based algorithm for learning the structure of the Bayesian network from data. Constraint based algorithms first determine some constraints in the form of conditional independence tests on data and then construct the skeleton of the graph that satisfies those constraints, meaning implied conditional independencies and V structures. In this paper we used the implementation `si.hiton.pc`, from the `bnlearn` package in the R programming language[2]. It uses the HITON algorithm[1] as the aid for feature selection and finding of Markov blankets, the sets of variables that make the variable independent from all others, for the PC algorithm in order to determine constraints that must be satisfied.

Before applying the PC algorithm, we also blacklisted certain edges from being included in the model that we knew made no sense to have in the network. Specifically, those were all the edges going from study habits to age and sex, which are background variables that can not be impacted by study habits. We also applied different values of the hyper parameter alpha, which controls the required strength of statistical

testing to determine whether the implied conditional independence holds. Other hyper parameters were left at default values.

2.2 Greedy hill-climbing algorithm

Score-based algorithms assign a score to each potential Bayesian Network and use a heuristic search strategy to select the optimal one. Hill-climbing is such an algorithm which uses a greedy-search strategy[5].

2.2.1 Akaike Information Criterion (AIC)

When fitting Bayesian models based on data, we want to create a model that balances accuracy and complexity. Fitting a model such that only accuracy is maximized will result in overfitting to the data. The formula for AIC is:

$$AIC = 2k - 2ln(L) \tag{1}$$

In this formula we prevent overfitting by penalizing the complexity of the model with a linear term that increases the AIC value with the number of independent network parameters k. The rest of the value is calculated with the maximized log-likelihood estimate L, i.e. $L = p(x|M)$. This term represents the accuracy of the model.

2.2.2 Bayesian information criterion (BIC)

Similarly to AIC the Bayesan Information criterion or BIC is a scoring function for optimal model selection. It, however, emphasizes more heavily on lower model complexity and thus introduces higher penalty term[3].

$$BIC = \log(N)k - 2\ln(L) \tag{2}$$

[4] where the term N is the sample size. $\log(N)$ acts like a none-constant regularizer unlike in 1. With its increase the score increases while in hill-climbing we are trying to find the network which minimize the score.

2.3 Comparing algorithms

For the comparison of algorithms we used hamming distance as a metric. It is the minimum number of alterations, meaning adding or removing an edge, needed to turn one network into the other. Lower hamming distance indicates more similar networks, and higher indicates more different networks. After applying hamming distance as a metric we also inspected the networks visually to comment on the differences between networks.

3 Results

3.1 PC-algorithm

The results for the PC-algorithm are presented in the figures in the appendix for the different values of hyper parameter alpha, which determines the required strength of the statistical test to determine conditional independence. Figure 11 is the result for alpha 0.01, Figure 22 for alpha 0.05 and figure 3 for alpha = 0.53.

We can determine from these results that alpha behaves as expected, stricter test requirements defined by lower alpha result in much sparser networks with many completely independent elements, as only the strongest statistical connections are learned. On the other hand, having a higher alpha results in a denser, more connected network.

We can also see that we get a much sparser network than we might expect, when given an alpha of the default size of 0.05. The reason for this is that our dataset has a lack of data when compared to the possible number of combinations of attributes, even after we have removed a lot of them in preprocessing. This makes many statistical tests much less reliable, as they require many samples for each possible combination of attributes to accurately determine the independencies, which simply is not available in our case. Increasing

the size of alpha to larger values is also not a good solution to the problem, as it reduces the reliability of tests and makes us more likely to include edges in the network that are result purely of random chance during sampling and are not statistically significant.

3.2 Hill climbing algorithm

The results for the hill climbing algorithm are presented in Figure 44 for AIC scoring and Figure 55 for BIC scoring.

We can see that the AIC result is much sparser, and has much more independent elements than the BIC result. The reason for this is that BIC penalises having more independent elements more harshly than the AIC, and therefore more edges are created in order to connect all variables together and optimise the score.

3.3 Algorithm comparison

For the purposes of this comparison we will use the result of the PC algorithm with the default alpha value of 0.05. We can see that it is somewhat similar to the result of the HC algorithm with AIC scoring. They result in sparse networks with many independent elements, however there are noticeably more edges in the PC algorithm network. Some of the edges are also shared between the networks. However, the result of the HC network with BIC scoring is very different from both of the other results. Due to greater penalisation of complexity, all the variables are connected into a single DAG and there are more edges because of it. It also shares some of the more statistically significant edges with the other networks.

With all things considered, and based on the domain knowledge, we would say that the HC algorithm with the BIC scoring performed the best in this task. We expect that the real network would be quite dense and not split into many independent elements, and penalisation of complexity results in a network that best matches those criteria.

3.4 Comparing algorithm results to our manual network

The manually constructed network is presented on Figure 66. We can see that it is much denser than any automatically constructed network presented here, even more than PC algorithm network with the very large value of alpha. In the manual construction of the network we heavily used domain knowledge which enabled us to define latent, bidirectional connections that are not present in the automatically constructed networks. We also used the polychoric correlation matrix and linear regression for testing conditional independencies, which tends to perform better when there is a lack of data as in our case.

The Hamming distances of several graph pairs are shown in Table 11. We can see that our manual graph is very different from the automatically generated graphs, while the latter are more similar to each other. Interesting to note is the high similarity of the networks from the HC algorithm and the PC algorithm with alpha of 0.05. The PC algorithm with alpha of 0.5 results in a less similar network compared to alphas of 0.05 and lower. The networks produced by these lower alphas are highly sparse, and were not considered for further analysis.

Table 1: Hamming distances between networks

Graph A	**Graph B**	**Distance**
Manual	BIC	33
Manual	AIC	35
Manual	PC-0.05	28
BIC	AIC	10
PC-0.05	BIC	7
PC-0.05	AIC	7
PC-0.05	PC-0.01	3
PC-0.05	PC-0.5	11
PC-0.01	PC-0.5	12

Overall, the manually constructed network seems better and more logical than automatically learned ones in this case, where there is a small number of variables and easily understandable domain. Even though we have demonstrated the various structure learning algorithms on this dataset, it is not the ideal example of a dataset where it would be sensible to apply the structure learning algorithms in the real world applications.

4 Discussion

In this study, we evaluated the performance of two bayesian inference algorithms, one of which is the constraint-based PC algorithm and one is score-based hill-climbing algorithm. We found that for the PC algorithm produced very sparse graphs unless the alpha parameter was set unreasonably high (0.5), which implies a high probability that the edges could have been created due to random chance. The HC algorithm with BIC seemed to produce a well-connected graph, more so than with the AIC scoring function. The BIC-graph is arguably better than the other graphs, as the AIC graph and PC graphs with reasonable alpha values have too many implied indepencencies.

However, we are not very satisfied with the implied directions of causality of some variables in the BIC-graph, especially for the arrows coming out of 'grade'. In the dataset, the grade data was collected after the results from the survey, so this variable should not affect the other variables.

For the inferred graphs, we had low expectations due to size of the dataset. As we saw from the results of the PC-algorithm, low values of alpha led to very sparse graphs, which is likely due to the small sample size. We did not expect the grade to be affected only by sex according to the graphs from the HC algorithm; we expected the effect of study habits to be much bigger than sex.

Structure learning algorithms, such as constraint-based and score-based algorithms used in this task, have limitations in that they only consider statistical relationships between data and not causal relationships. This can lead to inaccurate or nonsensical results. These algorithms are also not able to model latent variables. Additionally, these algorithms struggle when there is a lack of data. To address these limitations, we could explore different statistical tests and scoring methods to improve performance and consider using hybrid models that combine elements of constraint-based and score-based models.

References

[1] C.F. Aliferis, Ioannis Tsamardinos, and A Statnikov. "HITON: A Novel Markov Blanket Algorithm for Optimal Variable Selection". In: *AMIA ... Annual Symposium proceedings / AMIA Symposium. AMIA Symposium* 2003 (Feb. 2003), pp. 21–5.

[2] Constantin Aliferis et al. "Local Causal and Markov Blanket Induction for Causal Discovery and Feature Selection for Classification Part I: Algorithms and Empirical Evaluation". In: *Journal of Machine Learning Research* 11 (Jan. 2010), pp. 171–234. DOI: 10.1145/1756006.1756013.

[3] Zhifa Liu, Brandon Malone, and Changhe Yuan. "Empirical evaluation of scoring functions for Bayesian network model selection". In: *BMC bioinformatics*. Vol. 13. 15. BioMed Central. 2012, pp. 1–16.

[4] Gideon Schwarz. "Estimating the dimension of a model". In: *The annals of statistics* (1978), pp. 461–464.

[5] Marco Scutari. "Learning Bayesian Networks with the bnlearn R Package". In: ().

[6] Boran Sekeroglu, Kamil Dimililer, and Kubra Tuncal. "Student Performance Prediction and Classification Using Machine Learning Algorithms". In: Mar. 2019, pp. 7–11. ISBN: 978-1-4503-6267-2. DOI: 10.1145/3318396.3318419.

A Appendix

Table 2: Processed data

ID	Variable name	Type	Number of levels
1	Age	Ordered	(1: 18-21, 2: 22-25, 3: above 26)
2	Sex	Categorical	(1: female, 2: male)
17	Weekly study hours	Ordered	(1: None, 2: <5 hours, 3: 6-10 hours, 4: 11-20 hours, 5: more than 20 hours)
21	Impact of your projects/activities on your success	Ordered	(2: negative, 3: neutral, 1: positive)
22	Attendance to classes	Categorical	(1: always, 2: sometimes)
23	Preparation to midterm exams 1	Categorical	(1: alone, 0: not alone)
24	Preparation to midterm exams 2	Ordered	(3: never, 1: closest date to the exam, 2: regularly during the semester)
25	Taking notes in classes	Ordered	(1: never, 2: sometimes, 3: always)
26	Listening in classes	Ordered	(1: never, 2: sometimes, 3: always)
27	Discussion improves my interest and success in the course	Ordered	(1: never, 2: sometimes, 3: always)
28	Flip-classroom	Categorical	(1: useful, 0: not useful)
29	Cumulative grade point average in the last semester (/4.00)	Ordered	(1: <2.00, 2: 2.00-2.49, 3: 2.50-2.99, 4: 3.00-3.49, 5: above 3.49)
30	Expected Cumulative grade point average in the graduation (/4.00)	Ordered	(1: <2.00, 2: 2.00-2.49, 3: 2.50-2.99, 4: 3.00-3.49, 5: above 3.49)
O	Grade	Ordered	(0: Fail, 1: DD, 2: DC, 3: CC, 4: CB, 5: BB, 6: BA, 7: AA)

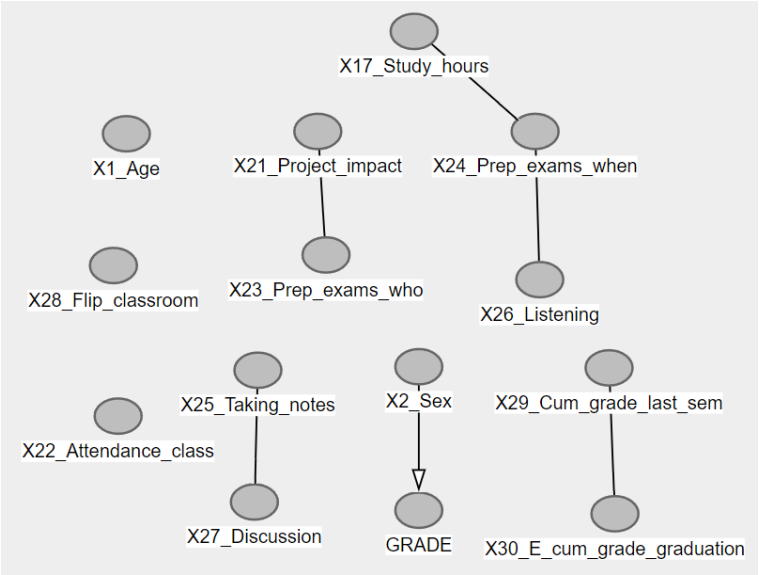

Figure 1: PC algorithm, alpha=0.01

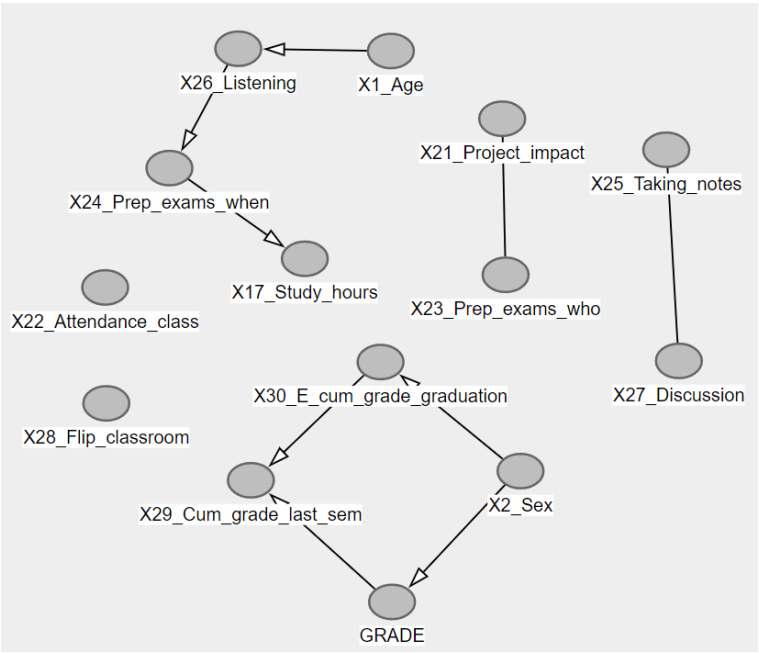

Figure 2: PC algorithm, alpha=0.05

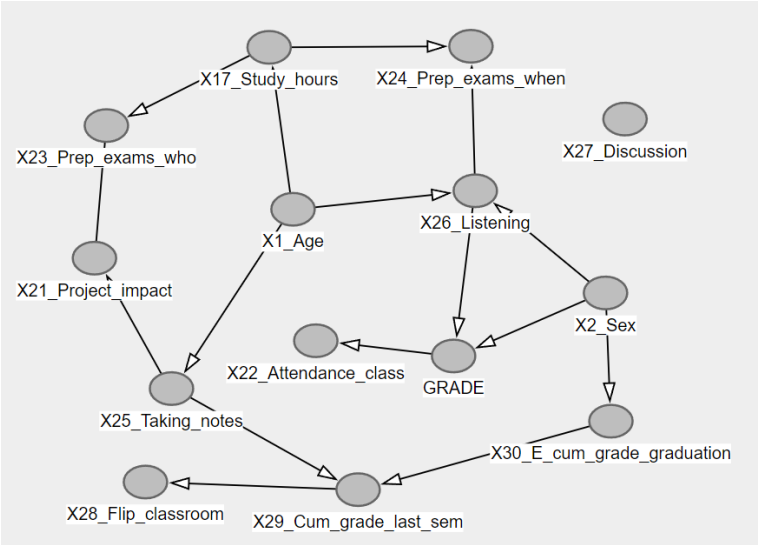

Figure 3: PC algorithm, alpha=0.5

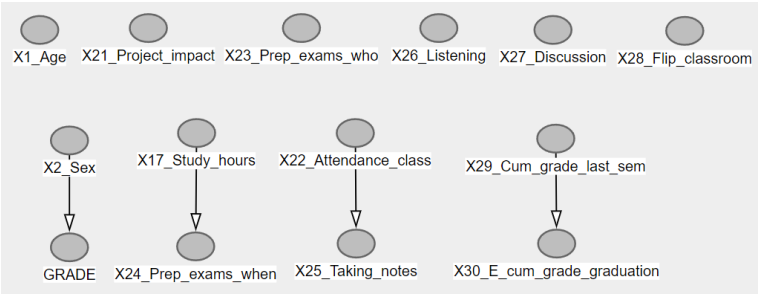

Figure 4: HC algorithm, AIC score

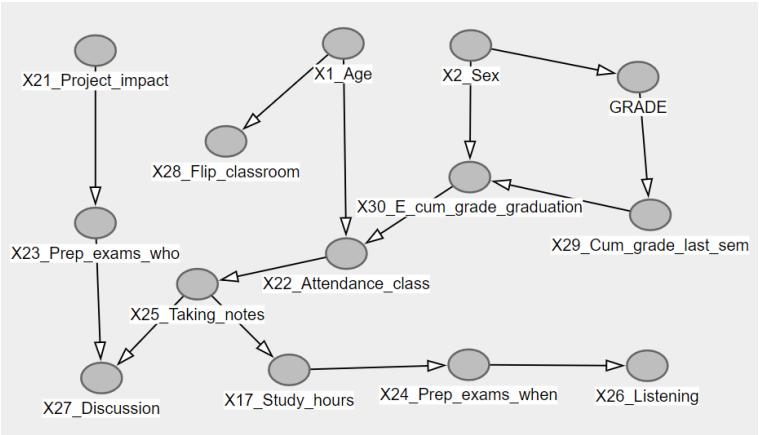

Figure 5: HC algorithm, BIC score

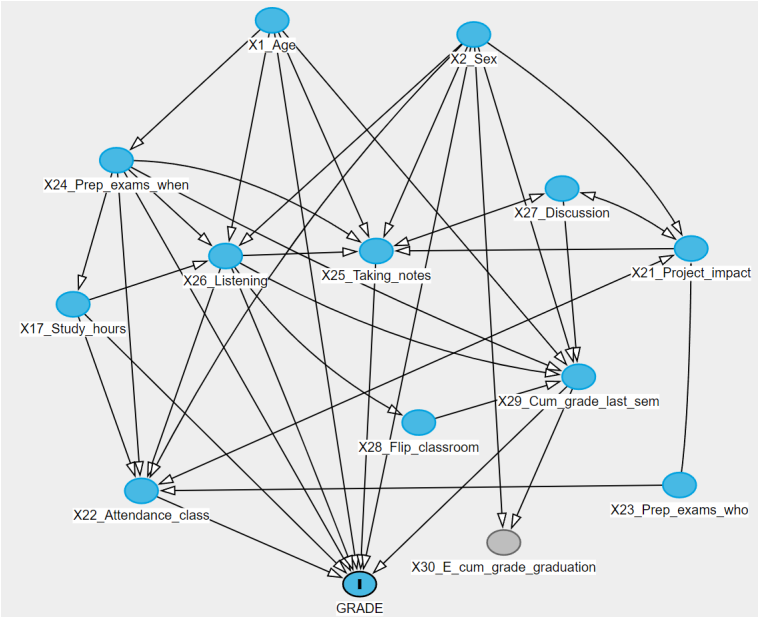

Figure 6: Our manual DAG