Causal Reasoning with Probabilistic Graphical Models

Probabilistic Graphical Models - Project Report - 15th March 2019

TECHNISCHE UNIVERSITÄT DARMSTADT

Moritz Willig 2527455; Johannes Czech 2422372

1 Introduction

The subject of causal reasoning is raising interest in today's research (Doshi-Velez and Kim (2017)). In 2018 the European Commission attaches great importance to the interpretability of artificial intelligence and AI-safety (*Interpretability in AI and its relation to fairness, transparency, reliability and trust - JRC Science Hub Communities - European Commission* (2018)). Probabilistic graphical models (PGM) have proved to be an effective method for expressing conditional dependencies among a set of random variables.

Figure 1: Judea Pearl

> *"The interpretation of DAGs as carriers of independence assumptions does not necessarily imply causation and will in fact be valid for any set of Markovian independencies along any ordering (not necessarily causal or chronological) of the variables. However, the patterns of independencies portrayed in a DAG are typical of causal organizations and some of these patterns can be given meaningful interpretation only in terms of causation."*
> – Judea Pearl (2013), Graphical models for probabilistic and causal reasoning

The application for PGM in the context for causal reasoning has been demonstrated by Pearl (2013) as well as Eichler and Didelez (2007). Bayesian Networks can be interpreted as causal networks if the directed edges have causal interpretation (Yin et al. (2008)). Finding the right edge direction in a Bayesian Network is a structure learning task. Causal reasoning can therefore be seen as a sub-field of Bayesian network structure learning. According to Pearl (2013) a causal network must not only be consistent with the conditional independence relationships but also be *stable*. He considers a Bayesian Network to be *stable* if alternative compatible structures are no longer compatible to slight fluctuations which are controlled in randomized experiments.

In this report we give a short overview of the currently available algorithms and frameworks to approach the task of structure learning. For this we use the LUCAS and CINA datasets for evaluation. Our code is available at `https://github.com/QueensGambit/PGM-Causal-Reasoning`

2 Problem Description

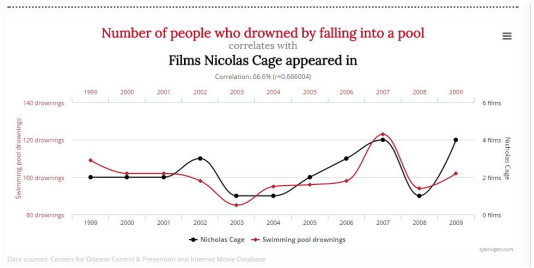

Figure 2: Correlation in Nicolas Cage appearance with swimming pool drowings

Causal discovery is a hard problem in exploratory data analysis. Its tasks is not only to find related attributes, but also to determine the direction of the causation. Even the simpler task of revealing correlating attributes from the data may lead to unexpected results. Thisapersists for pplying statistical methods: Given a sufficiently large set of features, there is a high chance for the existence of random correlations with high correlation coefficients between data attributes which in reality are unrelated. One also has to be careful to not confuse the cause and consequences and come to the wrong conclusions. An often cited example is the number incidents for people drowning in a pool correlating to the number of films the actor Nicolas Cage appeared in (*15 Insane Things That Correlate With Each Other* (2009)). There's a also the chance of finding a correlation but misinterpreting the direction: e. g. "the more firemen are sent, the more damage was done", "scholars who are getting tutored are getting worse grades than average". In the real world physical domain, causality often involves a time dependency where the cause happens prior to the effect. The information about this chronological ordering may not be present in the datasets. In contrast to this type of causality, causality may also be caused by the pure presence of a certain information. Agents which act in non fully observed environments may perform their actions solely on the state of some hidden variables. The pure exposure to certain bits of information to these systems, may cause the agent to perform an altered sequence of actions. This effect can for example be seen, when observing human game playing, where the own expectation about the opponents strength, and often leads to a more relaxed or serious type of game-play. Moreover, one should point out the difference between causality and correlation. Correlation is a precondition for causality. In contrast to correlation causality is an asymmetric relation. A single probability distribution can sometimes be represented with multiple different graph structures. One common example for this is the following three node graphs:

- Indirect causal effect: $X \longrightarrow Z \longrightarrow Y$
- Indirect evidential effect: $X \longleftarrow Z \longleftarrow Y$
- Common cause: $X \longleftarrow Z \longrightarrow Y$

All of these three graphs express the same independence relations and are therefore I-equivalent e. g. form a Markov equivalence class. In technical terms, finding the correct causal relationships involves identifying the right Markov blankets and immoralities in a set of random variables. Nonetheless, there's usually no 100% proof for causality in real-world examples. The problem of inferring general information from only a restricted set of previously observed data is discussed in general as the so called "problem of induction". We need to run explicit experiments or study via counter examples in order to verify a hypothesis.

3 Dataset Description

For analysing the problem of finding the correct PGM structure we evaluate our experiments on the LUCAS and CINA dataset.

3.1 **LU**ng **CA**ncer **S**imple set (**LUCAS**)

The LUCAS dataset was generated artificially by a predefined fixed Bayesian network. The dataset consists of 2000 samples in total with binary feature values. We chose this dataset because it enables a comparison of the generated structures with a ground truth network structure. Moreover, the size of the network is compact, interpretable and still trackable by a human. The conditional probabilites used to generate the artifical dataset can be examined at (*LUCAS* (2014)):

```
P(Anxiety=T)=0.64277
P(Peer Pressure=T)=0.32997
P(Smoking=T|Peer Pressure=F, Anxiety=F)=0.43118
P(Smoking=T|Peer Pressure=T, Anxiety=F)=0.74591
[...]
```

Code Listing 1: Excerpt of the first conditional probabilities of the Bayesian Network

4 **C**ensus **I**s **N**ot **A**dult dataset (**CINA**)

The CINA dataset is extracted from the 1994 Census database. For simplification we only use the discrete attributes and neglect the continuous ones because not all structure learning algorithms support a mixture of continuous and discrete data. If one wanted to make use of the full dataset, the continuous attributes could be discretized.

```
age: continuous.
workclass: Private, Self-emp-not-inc, ...
fnlwgt: continuous.
education: Bachelors, Some-college,...
education-num: continuous.
marital-status: Married-civ-spouse, Divorced,
occupation: Tech-support, Craft-repair, ...
relationship: Wife, Own-child, Husband, ...
race: White, Asian-Pac-Islander, ...
sex: Female, Male. ...
capital-gain: continuous.
capital-loss: continuous.
hours-per-week: continuous.
native-country: United-States, Cambodia,...
England, Puerto-Rico, Canada, Germany,...
income: >50K, <=50K
```

Code Listing 2: Overview of different attributes in the CINA dataset *CINA* (2014)

5 Evaluated Frameworks and Tools

For this report we considered the following set of tools and frameworks:

Table 1: List of frameworks

Name	Description	Source-Code Link	Structure Learning
SPFlow	An Easy and Extensible Library for Sum-Product Networks	https://github.com/SPFlow/SPFlow	✓
BNFinder	Tool for learning bayesian networks	https://github.com/sysbio-vo/bnfinder	✓
bnlearn	An R package for Bayesian network learning and inference	http://www.bnlearn.com/	✓
BayesSpy	Bayesian Python	https://github.com/bayespy/bayespy	✗
OpenGM	A C++ template library for discrete factor graph models and distributive operations	http://hciweb2.iwr.uni-heidelberg.de/opengm/	✗

5.1 SPFlow

SPFlow (Molina et al. (2019)) is an open-source python framework developed by Alejandro Molina et al. published under Apache License, Version 2.0. It allows among other thing the generation of sum product networks by a given dataset. We use `learn_mspn()` with `min_instances_slice=200` and `threshold=0.4` to generate a SPN-structure on the loaded LUCAS training data container, figure 4. Despite that sum product networks are suitable for running conditional queries and scale well with big datasets, we found the resulting network graphs hardly interpretable and not suitable for examining causal relationships using the resulting sum product network structure.

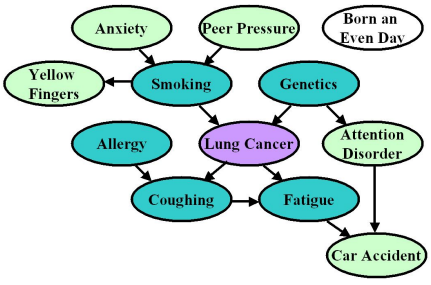

Figure 3: BN structure of LUCAS dataset, *LUCAS* (2014)

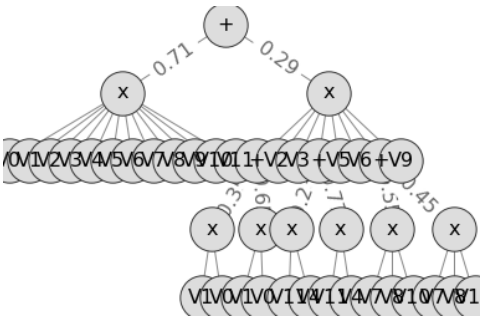

Figure 4: Sum product network graph on the LUCAS dataset

5.2 BNFinder

BNFinder (Frolova and Wilczyski (2018)) is a tool for finding bayesian network strutures written by Alina Frolova and Bartek Wilczyski2 in python 2.X. and released under the GNU GPL 2.0 license. It is primarily used in the medical domain for gene regulatory network inference and makes use of distributed and parallel computing. The tool relies on prior knowledge by defining a set of `#parents` or `#regulators`.

```
#regulators     Anxiety  Peer_Pressure    Genetics        Born_an_Even_Day
conditions      EXP0     EXP1     EXP2  [...]
Smoking  0      0        1        0  [...]
```

Code Listing 3: Excerpt from `lucas0_train_bnfinder.txt`

Using BNFinder on the LUCAS dataset with the specificied `#regulators` as shown in listing 3, we get the following list of positive correlations:

```
Anxiety +       Smoking
Anxiety +       Yellow_Fingers
Peer_Pressure   +       Smoking
Genetics        +       Attention_Disorder
Genetics        +       Lung_cancer
Car_Accident    +       Attention_Disorder
Car_Accident    +       Fatigue
Car_Accident    +       Lung_cancer
```

Code Listing 4: `output1.sif`

Most of the correlations are reasonable, but we found the functionality of BNFinder too specific in the aspect of the need of prior knowledge.

5.3 bnlearn

bnlearn (Scutari (2009)) is an R package by Marco Scutari and Robert Ness, which is still in continuous development since 2007. It provides constraint-based, score-based and hybrid structure learning algorithms.

5.4 Structure Learning Algorithms

Algorithms for structure learning can be divided into two common groups. The first group, so called **score based** algorithms, generate a set of potential network structures and evaluate them on the data, based on a predefined score. The other group of **constrained based** algorithms tries to construct BNs directly from the data, considering observed correlations and independency constraints.

5.4.1 Constrained Based

A number of constrained based algorithms exist. These algorithms all make use of the given data to infer the network structure. In contrast to score based algorithms the data is not only used to evaluate the quality of a given network structure, but it is considered during the construction of the network to directly incorporate independency constraints. We looked at the following algorithms:

- Grow-Shrink (Margaritis (2003))
- Incremental Association (Tsamardinos, Aliferis and Statnikov (2003))
- Fast Incremental Association (Yaramakala and Margaritis (2005))
- Interleaved Incremental Association (Tsamardinos et al. (2003))
- Max-Min Parents and Children (Tsamardinos, Brown and Aliferis (2006))

To give some intuition into this kind of algorithms, we briefly explain the Grow-Shrink algorithm in more detail.

5.4.2 Grow Shrink Algorithm

As with most structure learning algorithms, this algorithm tries to find the structure of a network given only the dataset. It was first described in Margaritis (2003).

In the first step of GS tries to find the markov blanket for each variable V. It keeps track of a list of nodes S that potentially belong to the markov blanket. The algorithm initially starts with an empty candidate list. In the **growing phase** GS iterates through all nodes. Each node that is dependent on V given the current S, is added to S. It is therefore considered a potential candidate to the markov blanket of V. In this step it is possible for the algorithm to add other variables than those that belong to the 'true' markov blanket. While some variables of the true markov blanket are not yet included in S, it is possible that others, outside the markov blanket are still dependent on V. For now, these variables still get added to S. To remove these unnecessary variables, a consecutive **shrinking phase** is performed. In this phase, all candidates Y in S that are independent of the variable considering the remaining markov blanket $S - Y$ get removed. After this step the candidate list only contains variables of the true markov blanket for V. The remaining steps of the GS algorithm then build a graph structure out of the discovered markov blanket, test for the correct orientation of the edges and remove any cycles from it, to finally recover the final Bayesian network.

5.5 Network Scores

For the other type of algorithm, hill-climbing or other general purpose heuristic search strategies, such as tabu search or simulated annealing, are used. These methods typically require a score to rank the proposed network structures. The most common scores are:

- (Log-)Likelihood (Witten, Frank, Hall and Pal (2016))

$$\log \mathrm{L} = \mathrm{M} \sum_i (\mathrm{I}(x_i; Pa_i) - \mathrm{H}(x_i)) \tag{1}$$

- **A**kaike **I**nformation **C**riterion (**AIC**) (Akaike (1974))

$$\mathrm{AIC} = \log \mathrm{L}(X_1, ..., X_v) - d \tag{2}$$

- **B**ayesian **I**nformation **C**riterion (**BIC**) (Schwarz (1978))

$$\mathrm{BIC} = \log \mathrm{L}(X_1, ..., X_v) - \frac{d}{2} \log n \tag{3}$$

- **B**ayesian **D**irichlet **E**quivalent Score (**BDE**) (Heckerman, Geiger and Chickering (1995))
- **K2** score (Cooper and Herskovits (1992))

The goal behind all the these scores is to find a good weighting between coherence of the network to the data and the complexity e. g. number of edges of the BN.

6 Result Comparison on the LUCAS Dataset

As can be seen in figure 5 and 6, Incremental Association, Fast Incremental Association and Interleaved Incremental Association were able to fully recover the ground truth bayesian network structure with all correct edge directions. Only the Grow-Shrink algorithm is missing out an edge.

Figures 7 and 8 show a worse performance for the score based methods. Out of the three scores, BIC seems to performs best. No edge is left out, and only two additional edges are added. As the log-likelihood score does not punish the complexity of a model, it allows all edges to get added. Adding edges always enables the network to represent a larger set of relationships. However, getting an always fully-connected network, does not help us with finding the real underlying correlations and causal structures in the data.

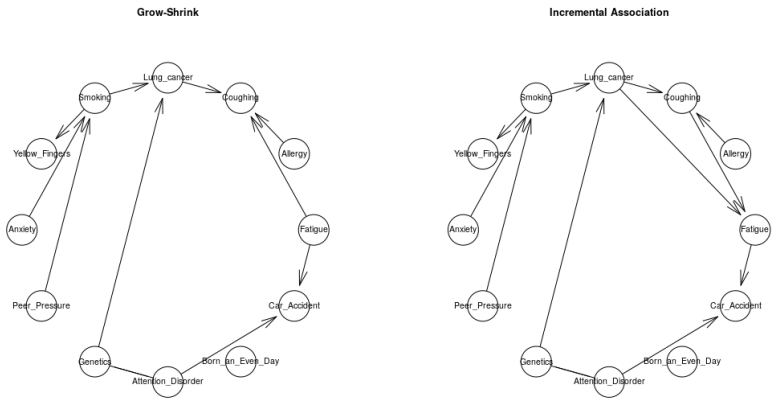

Figure 5: Comparison between Grow-Shrink and Incremental Association

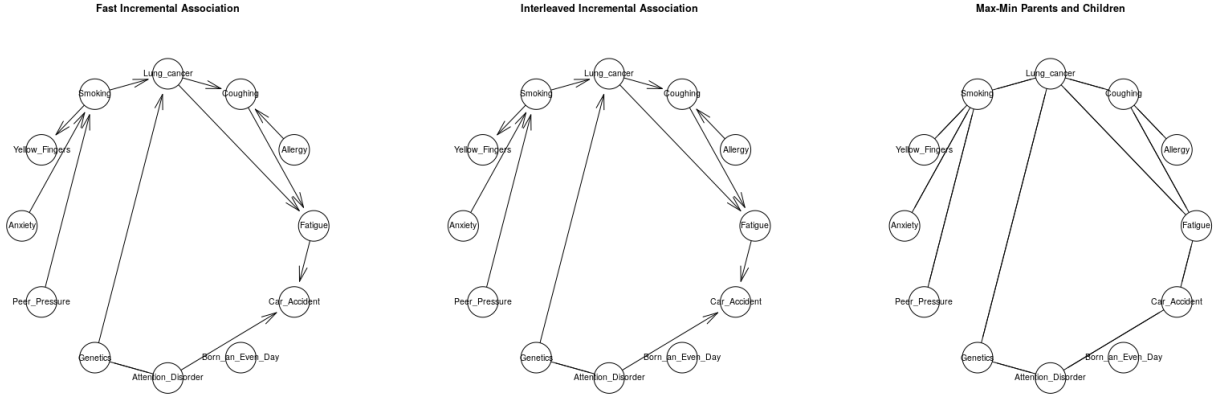

Figure 6: Comparison between Fast Incremental Association, Interleaved Incremental Association and Max-Min Parents and Children

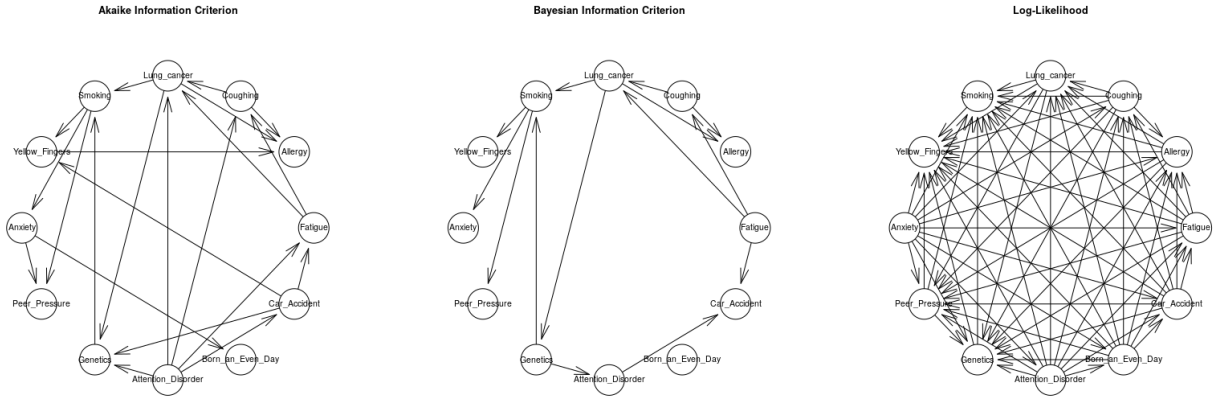

Figure 7: Comparison of Hill-Climbing using AIC, BIC and Log-Likelihood

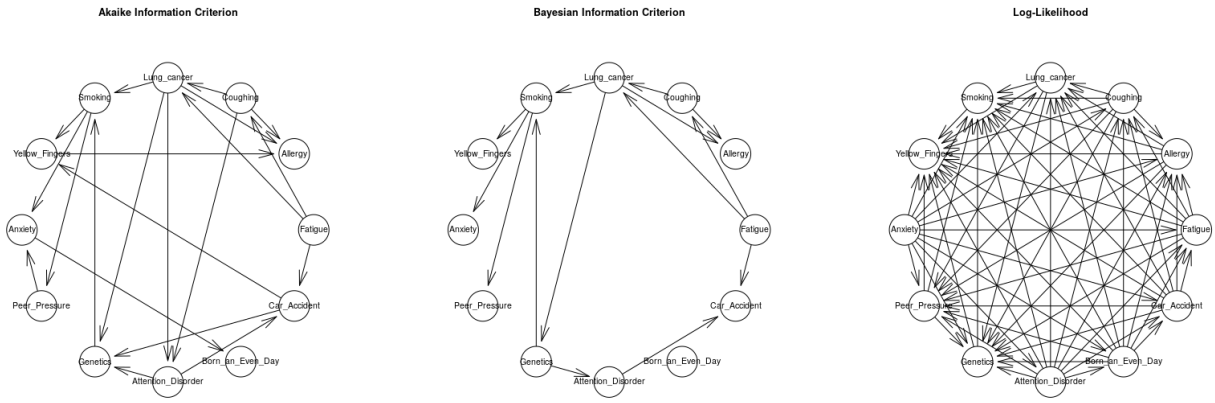

Figure 8: Comparison of Tabu-Search using AIC, BIC and Log-Likelihood

7 Result Comparison on the CINA Dataset

Since there exists no ground truth for the CINA dataset we have to rely on our intuition. Linking variables like marital status to relationship, or education and income seems reasonable. For figures 9, 10, 11, 12, we see that most of the constrained based methods seem to produce networks with reasonable connections. Like we have seen on the LUCAS dataset, the score based methods tend to create more edges than necessary.

When considering our goal of extracting causal information, we recognize that out of the score based methods only AIC and log-likelihood managed to extract the education and income correlation. We believe that this is a typical example of a chronological cause-consequence pair. We assume that most people get least a minimum level of school education before starting to work and having their own income. Therefore, the level of education one receives should affect his or her income later on. However, in all of the three cases where the correlation is discovered by the score-based networks, it is represented as an "income affects education"-edge.

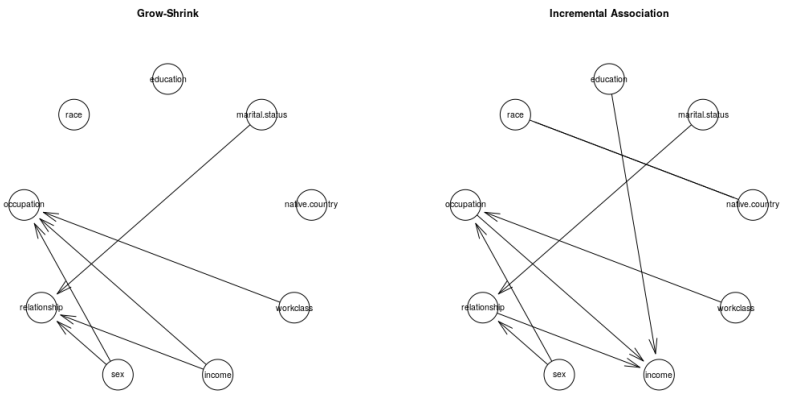

Figure 9: Comparison between Grow-Shrink and Incremental Association

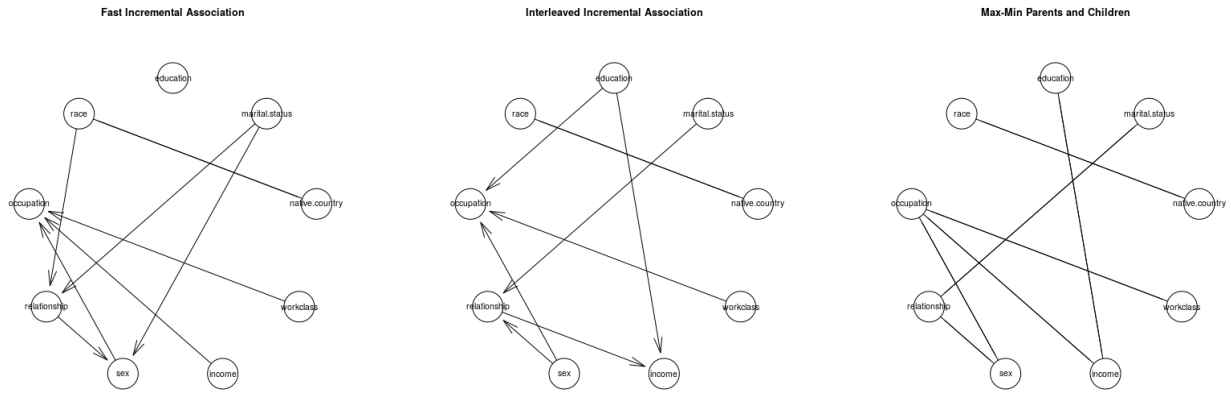

Figure 10: Comparison between Fast Incremental Association, Interleaved Incremental Association and Max-Min Parents and Children

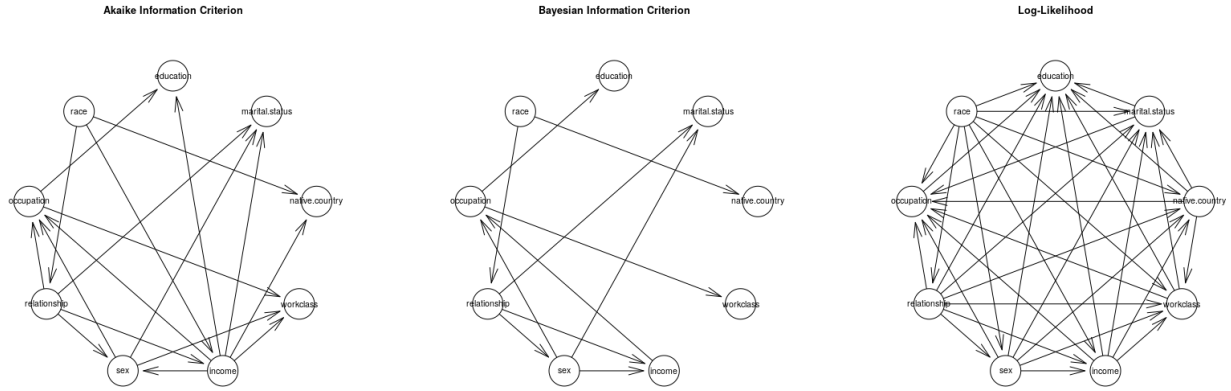

Figure 11: Comparison of Hill-Climbing using AIC, BIC and Log-Likelihood

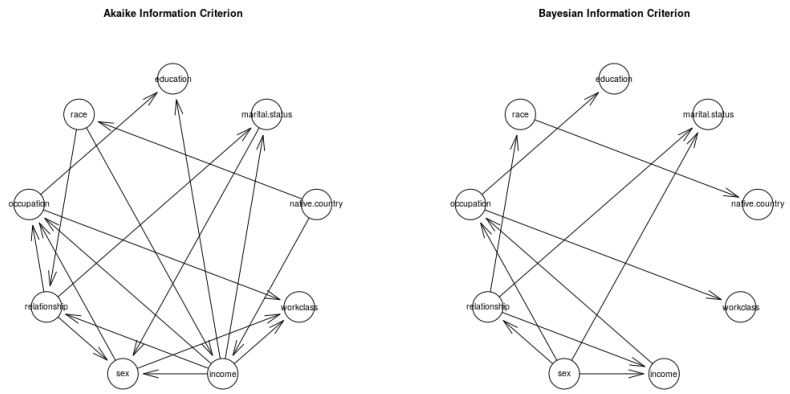

Figure 12: Comparison of Tabu-Search using AIC, BIC

8 Conclusion

From the examples we have looked at, we conclude that constrained based algorithms, such as Incremental Association or Grow-Shrink, seem to outperform score based methods. Constrained based algorithms seem to create better networks, representing the underlying correlations and independencies of the data. Additionally, better results for the direction of causal relations is achieved for constraint based methods. The score based algorithms struggle to find the right balance between the cost of further adding or removing edges and optimizing the cost of the network on the data. Moreover, the direction of the edges seem to be unstable or even consistently inverted.

From all the frameworks we consider bnlearn to be a good out of the box framework for learning structure in general networks. While the other toolboxes are designed to perform tasks in their specific domain. In out case they lack flexibility to apply them directly to general purpose structure learning.

On the LUCAS dataset we experienced a high correspondance with the actual ground truth network, but the varying results on the CINA suggest that none of the tested algorithms manage to outperform the others consistently. We observed that edges, constructed by multiple algorithms, are more likely to also be present in the ground truth. Other than using a simple ensemble methods with majority voting, we recommend further empirical studies to reveal the strengths, weaknesses or systematic errors made by the individual algorithms, that may lead to a more robust prediction of the BN structure.

References

15 insane things that correlate with each other. (2009). Retrieved 2019-02-25, from `http://tylervigen.com/spurious-correlations`

Akaike, H. (1974, December). A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, *19*(6), 716-723. doi: 10.1109/TAC.1974.1100705

CINA. (2014). Retrieved 2018-11-01, from `http://www.causality.inf.ethz.ch/data/CINA.html`

Cooper, G. F. & Herskovits, E. (1992). A bayesian method for the induction of probabilistic networks from data. *Machine learning*, *9*(4), 309–347.

Doshi-Velez, F. & Kim, B. (2017). Towards a rigorous science of interpretable machine learning. Retrieved 2018-11-01, from `http://arxiv.org/abs/1702.08608`

Eichler, M. & Didelez, V. (2007). Causal reasoning in graphical time series models. , 8.

Frolova, A. & Wilczyski, B. (2018, October). Distributed bayesian networks reconstruction on the whole genome scale. *PeerJ*, *6*, e5692. Retrieved from `https://doi.org/10.7717/peerj.5692` doi: 10.7717/peerj.5692

Heckerman, D., Geiger, D. & Chickering, D. M. (1995). Learning bayesian networks: The combination of knowledge and statistical data. *Machine learning*, *20*(3), 197–243.

Interpretability in AI and its relation to fairness, transparency, reliability and trust - JRC science hub communities - european commission. (2018). Retrieved 2018-11-01, from `https://ec.europa.eu/jrc/communities/community/humaint/article/interpretability-ai-and-its-relation-fairness-transparency-reliability-and`

LUCAS. (2014). Retrieved 2019-02-23, from http://www.causality.inf.ethz.ch/data/LUCAS.html

Margaritis, D. (2003). Learning bayesian network model structure from data.

Molina, A., Vergari, A., Stelzner, K., Peharz, R., Subramani, P., Mauro, N. D., ... Kersting, K. (2019). *Spflow: An easy and extensible library for deep probabilistic learning using sum-product networks.*

Pearl, J. (2013). Graphical models for probabilistic and causal reasoning. , 29.

Schwarz, G. (1978). Estimating the dimension of a model. *Ann. Statist.*, *6*(2), 461–464.

Scutari, M. (2009). Learning bayesian networks with the bnlearn r package. Retrieved 2019-02-23, from http://arxiv.org/abs/0908.3817

Tsamardinos, I., Aliferis, C. F. & Statnikov, A. (2003). Algorithms for Large Scale Markov Blanket Discovery. , 5.

Tsamardinos, I., Brown, L. E. & Aliferis, C. F. (2006). The max-min hill-climbing bayesian network structure learning algorithm. *Machine learning*, *65*(1), 31–78.

Witten, I. H., Frank, E., Hall, M. A. & Pal, C. J. (2016). *Data mining: Practical machine learning tools and techniques*. Morgan Kaufmann.

Yaramakala, S. & Margaritis, D. (2005). Speculative markov blanket discovery for optimal feature selection. In *Fifth ieee international conference on data mining (icdm'05)* (pp. 4–pp).

Yin, J., Zhou, Y., Wang, C., He, P., Zheng, C. & Geng, Z. (2008). Partial orientation and local structural learning of causal networks for prediction. , 13.