Assignment 2

Learning Bayesian Networks from Data

Handout for the *Probabilistic Graphical Models* lecture, December 12th, 2017

Johannes Textor, Perry Groot, Marcos Bueno

Objectives of This Exercise

1. Apply a structure learning algorithm to a real-world dataset.
2. Appreciate the possibilities and limitations of structure learning.

In this assignment, you will apply and compare two algorithms that attempt to infer the structure of a graphical model entirely from data. The choice of the algorithms is up to you; a list of suggestions is given at the end of this document, but you can also use a different algorithm that is not listed here (if you intend to do this, please contact me beforehand). The only important requirement is that your algorithms should be based on two substantially different approaches – for example, you should not compare two different implementations of the PC algorithm.

Ideally, you should use your algorithm on the data from your first assignment. If you would prefer not to use this data, please contact me as soon as possible, so I can provide you with an alternative data set. (You should contact me **before** you choose your algorithm, since not all algorithms are applicable to all data sets.)

Tasks

- Choose two algorithms for learning a graphical model (e.g., a DAG or CPDAG). Make sure that you choose an algorithm that is actually applicable to your data.
- Read the scientific literature (e.g., a paper) that describes these algorithms in detail.
- Design an evaluation metric. That is, define one or several quantitative measure(s) that can be used to compare the structure of the resulting networks to each other. Focus your evaluation on the actual graph structure, not on the parameters of the learned probability distributions.
- Apply these two algorithms to your dataset from Assignment 1. Compare (using your metric) the results with each other, and to your manually constructed network.
- For at least one parameter of each algorithm, perform an analysis of how the results change with this parameter, and give a recommendation which value should be used in your setting.

Report

Write an overall report on your findings. This report should be structured in the usual way of reporting an empirical scientific study, that is, it should contain the following sections:

1. **Introduction:**
   - What motivated your choice of algorithms? Specifically, which of its aspects seemed particularly promising or practical to you?
   - Where in the literature are these algorithms first described?
   - How do the algorithms work? Which parameters do they have and how does each parameter influence the results? Are there any guidelines as to how these parameters should be chosen?
   - Give a brief description of your dataset to make this a self-contained report.
2. **Methods:**
   - Describe how the algorithms are implemented (e.g., R package, python package, ...).
   - Describe any preprocessing steps that had to be performed to input the data into the algorithms.
   - Describe how you compute your distance measure between graphs.

3. **Results:**
   - Show the results of the structure learning, and show how the results change if (at least) one of the parameters changes.
   - Compare the results of the two algorithms with each other.
   - Compare the results of the two algorithms to your hand-constructed network from Assignment 1 (or, if you are not using these data, a hand-constructed network that we will provide to you).
4. **Discussion:**
   - Summarize your results, and comment on whether they match your expectations. Specifically, mention which of the two algorithms performed better (under which circumstances).
   - If you identify any limitations of the algorithms you've used, can you suggest improvements? Is there any existing work that tries to improve on these algorithms?

Suggested Algorithms

Below is a list of structure learning algorithm implementations that you can choose from or use as a starting point for your own research.

- The R package `bnlearn`; particularly, the functions `hc`, `tabu`, and `si.hiton.pc`.
- The R package `pcalg`; particularly, the functions `pc` and `rfci`.
- The R package `lavaan`. While this package does not contain a structure learning algorithm, you can easily use it to create your own by building on the function `modindices`.
- The Python library `pgmpy` (`http://pgmpy.org`), see the function `est.estimate_skeleton` and the function `est.skeleton_to_pdag`.
- The software "GOBNILP", see `https://www.cs.york.ac.uk/aig/sw/gobnilp/`. Apparently a Python wrapper is available at `https://github.com/ncullen93/pyBN`.
- The TETRAD project at `http://www.phil.cmu.edu/projects/tetrad/`, a Java software that implements various algorithms.
- The Bayes net toolbox for Matlab (`https://github.com/bayesnet/bnt`) contains an implementation of the PC algorithm.