Exercise: Bayesian Nets
Data Science Specialization (Spring 2025)

Jens Classen
IMT, Roskilde University
jens-classen.net
classen@ruc.dk

26.03.2025

In this exercise, we want to explore Bayesian networks. In Moodle, you find a file called `transport.csv` with an (artificial) dataset from a survey, using the following features/variables:

- `Age`: the person's age, values: `young`, `middle`, `old`
- `Education`: the person's education level, values: `low`, `high`
- `Income`: the person's income level, values: `low`, `medium`, `high`
- `Residency`: size of the town where the person resides, values: `small`, `big`
- `Transport`: preferred means of transport, values: `car`, `train`

The task is to find the Bayesian network that best fits the data using *parameter learning*. For this purpose, try out different initial structures for the net, and run the algorithm for different sizes of training sets.

1 Installing `pgmpy`

Here we will use the `pgmpy` library for probabilistic graphical models in Python. It can can installed in Anaconda using the command

```
conda install -c ankurankan pgmpy
```

Documentation can be found at `https://pgmpy.org/`.

Tip:

- Note that in the current version, `pgmpy` by default does not print the entire CPD if it does not fit into the width of the terminal. A workaround can be found here: `https://stackoverflow.com/a/74350759`

2 Learning Bayesian Networks

The `pgmpy` documentation has a number of tutorials in the form of Jupyter notebooks. In particular, a tutorial on learning (both structure and parameters of) Bayesian networks from data can be found here: `https://pgmpy.org/detailed_notebooks/10.%20Learning%20Bayesian%20Networks%20from%20Data.html`

3 Evaluating Models

While it is possible to evaluate a Bayes Net in a similar fashion as other classifiers, note that *prediction accuracy* might not be the best metric to use due to the inherent stochastic nature of the domain we are using. As a simple example, if we only have a single variable *Flip* denoting the outcome of flipping a fair coin, we might have 50.1% of the examples with value *heads* and 49.9% with *tails*. A maximum a posterio predictor will always choose *heads*, and hence only show an accuracy of around 50.1%.

It is better to check whether the *probability distribution* represented by the network matches the one in the data as closely as possible. Possible metrics are the *correlation score* and the *log likelihood score*. You can read about them under `https://pgmpy.org/metrics/metrics.html`.

4 Bonus Exercise

Find a larger datset from an application that is of interest to you and apply Bayesian Learning to it. You may consider evaluating parameter learning wrt a number of hand-tailored structures, analyzing the feasibility of structure learning, comparing the performance of a Bayes net against a naive Bayes classifier, and more. You may also consider to include continous features.

Possible candidates are medical applications with known (or assumed) causal relations among features, demographical data, etc.