# Evaluating methods for learning graphical structures from data

## 1 Description

One important application of unsupervised learning algorithms is to learn graphical structures from data. Several algorithms have been proposed with this purpose [1, 2, 3, 4, 5]. Some datasets encode information about the interactions between the variables. For example, an analysis of a dataset can reveal that two genes have a correlated protein expression. Usually, in a graphical structure that represent these relationships, each variable is represented by a node and an edge (or arc) is used to represent whether the two variables are related. Alternatively, graphical models can be use to capture conditional independence relatioships between the variables. There are specific methods for datasets of continuous, discrete, and mixed values.

One of the problems with methods that learn graphical models is how to validate their results since some methods tend to miss edges corresponding to existing relationships in the data, and other methods tend to introduce spurious or false edges. A good method should be sensitive to capture all relevant edges, and specific to avoid capturing false edges.

## 2 Objectives

The goal of the project is to compare the Python implementations of 3 or more methods for learning the graphical structure from data on a set of 5 or more datasets. The students should: 1) Identify the implementations available in Python for learning graphical structures from data. 2) Identify which datasets are relevant for inferring the graphical structure from the data. 3) Prepare the experiment. 4) Propose the validation framework.

As in other projects, a report should describe the characteristics of the design, implementation, and results. A Jupyter notebook should include calls to the implemented function that illustrate the way it works.

## 3 Suggestions

- See example for learning graph structure from data in `http://scikit-learn.org/stable/auto_examples/applications/plot_stock_market.html`.
- Read paper [2] to identify candidate datasets.
- Inspect work on graphical models [1, 3, 4, 5]
- Implementations can use any Python library, in particularly, the *libpgm* for learning the structure of Bayesian networks.
- For validation, consider the independent evaluation of the sensitivity and specificity of the learning method for capturing the original pairwise interactions (or original edges).

## References


[1] E.M. Airoldi. Getting started in probabilistic graphical models. *PLoS Computational Biology*, 3(12):e252, 2007.

[2] Eugene Belilovsky, Kyle Kastner, Gael Varoquaux, and Matthew B Blaschko. Learning to discover sparse graphical models. In *Workshop track. ICLR 2017*, pages 1–13, 2017.

[3] W. Buntine. A guide to the literature on learning graphical models. *IEEE Transactions on Knowledge and Data Engineering*, 8:195–210, 1996.

[4] Greg F. Cooper and Edward Herskovits. A Bayesian method for the induction of probabilistic networks from data. *Machine Learning*, 9:309–347, 1992.

[5] David Heckerman, Dan Geiger, and David M. Chickering. Learning Bayesian networks: The combination of knowledge and statistical data. *Machine Learning*, 20:197–243, 1995.