A comparison of methods for structural learning of Bayesian networks from data

1 Description

Bayesian networks (BNs) [1, 2, 3] are probabilistic graphical models that serve to represent probabilistic dependencies between variables in uncertain domains. They are one of the most popular approaches in machine learning and have been applied to a large number of real-world problems [4, 5, 6, 7, 8]. Despite the popularity of BNs, the implementation of methods for learning and inferring these graphical models from data is not straightforward and can be computationally costly [9, 10, 11]. Different learning techniques exhibit different behaviors.

2 Objectives

The goal of the project is, for a given dataset of binary values, learn the Bayesian structure and parameters using two different types of learning methods: 1) Score-based Structure Learning. 2) Conditional Independence Tests. A Python program will be implemented that receives a dataset of binary vectors that satisfy a number of probabilistic dependencies. From this input, the program will output two Bayesian Networks representing the dependencies encoded in the data. Each BN will be learned using a different method.

The student should: 1) Implement the program that learn the BNs (see suggestions below). 2) Generate test sets satisfying different independence conditional relationships between the variables to validate the program. 3) Visualize the BNs structures learned.

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

- Read the paper "Bayesian networks without tears" `https://www.aaai.org/ojs/index.php/aimagazine/article/download/918/836`
- Try the pgmpy notebooks introducing BNs and methods to learn them from data: `https://github.com/pgmpy/pgmpy_notebook/tree/master/notebooks`.
- Visit the Bayesian Network Repository for other examples of Bayesian networks: `http://www.cs.huji.ac.il/~galel/Repository/#Networkformats`
- Implementations can use any other Python library.

References

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

[2] K. Murphy. *Machine Learning: A Probabilistic Perspective*. MIT Press, 2012.

[3] J. Pearl. *Causality: Models, Reasoning and Inference*. Cambridge University Press, 2000.

[4] S.K. Andersen, K.G. Olesen, F.V. Jensen, and F. Jensen. HUGIN– a shell for building Bayesian belief universes for expert systems. In *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*, volume 2, pages 1080–1085, 1989.

[5] Amir Hossein Ashouri, Giovanni Mariani, Gianluca Palermo, and Cristina Silvano. A Bayesian network approach for compiler auto-tuning for embedded processors. In *Embedded Systems for Real-time Multimedia (ESTIMedia), 2014 IEEE 12th Symposium on*, pages 90–97. IEEE, 2014.

[6] C. Echegoyen, J. A. Lozano, R. Santana, and P. Larrañaga. Exact Bayesian network learning in estimation of distribution algorithms. In *Proceedings of the 2007 Congress on Evolutionary Computation CEC-2007*, pages 1051–1058. IEEE Press, 2007.

[7] Ramon Etxeberria and Pedro Larrañaga. Global optimization using Bayesian networks. In A. Ochoa, M. R. Soto, and R. Santana, editors, *Proceedings of the Second Symposium on Artificial Intelligence (CIMAF-99)*, pages 332–339. Editorial Academia. Havana, Cuba, 1999.

[8] P. H. Lee and H. Shatkay. Bntagger: improved tagging SNP selection using Bayesian networks. *Bioinformatics*, 22(14):e211–219, 2006.

[9] David M. Chickering, Dan Geiger, and David Heckerman. Learning Bayesian networks is NP-hard. Technical Report MSR-TR-94-17, Microsoft Research, Redmond, WA, 1994.

[10] David M. Chickering, David Heckerman, and Christopher Meek. A Bayesian approach to learning Bayesian networks with local structure. Technical Report MSR-TR-97-07, Microsoft Research, Redmond, WA, August 1994.

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