bnstruct: R Package for Bayesian Network Structure Learning in the Presence of Missing Data

Alberto Franzin AFRANZIN@ULB.AC.BE
IRIDIA, Université Libre de Bruxelles, 1050 Brussels, Belgium

Francesco Sambo FRANCESCO.SAMBO@UNIPD.IT
Barbara Di Camillo DICAMILL@DEI.UNIPD.IT
Department of Information Engineering, University of Padova, 35131 Padova, Italy

Keywords: Bayesian Network, Structure Learning, Missing Data

Abstract

Bayesian Networks are Probabilistic Graphical Models that represent the set of (conditional) probabilistic relationships in a set of random variables. We introduce `bnstruct`, an open source R package for learning the structure and the parameters of a Bayesian Network also in presence of missing data, and for computing inference on the network.

An extended version of this abstract has been submitted to the *Bioinformatics* journal.

1. Introduction

Bayesian Networks (BN) are Probabilistic Graphical Models that can represent the set of causality relationships in a set of random variables (Pearl, 1988; Koller & Friedman, 2009). A BN can be fully described as $B = (X, G, \Theta)$, where X is the set of random variables, G is a Directed Acyclic Graph (DAG) that forms the *structure* of the network, that is, the graph that represents the conditional independencies between the variables, and Θ is the set of *parameters* that specify the conditional probability distributions of the variables. BNs are an intuitive and powerful tool, and are used in many contexts, from biology to computer vision. BNs allow to perform *inference*, that is, estimating how the modified knowledge about the state of some variables propagates throughout the rest of the network.

Many applications require to *learn* the structure G of a BN that may have generated a given dataset D; subsequently, the set Θ of parameters can be estimated from D, according to G. The *structure learning* problem is NP-complete, and several exact and heuristic algorithms have been proposed for it.

In many contexts, the dataset D is also not complete and might miss some observations, due to loss of observations, values out of the measurement thresholds, or impossibility of taking measurement. In this case, the possibilities are: to use only the observations with no missing values (*complete cases* analysis), the observations without missing values for a certain subset of variables (*available cases* analysis), to guess the value of the missing observations starting from the available data (*imputation*), or to use ad-hoc algorithms.

Althought some packages for BN learning already exist, to the best of our knowledge there are no available open source implementations of learning and inference algorithms that can handle missing data. Both `bnlearn` (Scutari, 2010), `CGBayesNets` (McGeachie et al., 2014) and `BNT` (Murphy, 2001) perform learning and inference, but do not deal with missing values; `MoTBFs` (Pérez-Bernabé & Salmerón, 2015) allows only complete case analysis; `PEBL` (Shah & Woolf, 2009) can perform learning with missing values, but cannot perform inference.

In this work we introduce an open source R package, `bnstruct` providing methods for: 1) learning the structure and the parameters of a network from a dataset that can contain missing data, 2) learning a robust network structure using bootstrap resampling, 3) performing inference on a network. `bnstruct` works with both discrete and continuous variables; the latter get discretized. `bnstruct` is freely available on CRAN, with a rich documentation and a set of examples.

Appearing in *Proceedings of Benelearn 2016*. Copyright 2016 by the author(s)/owner(s).

2. Methods

Data: `bnstruct` requires a dataset and associated metadata for each variable (name, cardinality/discretization level, discreteness). For datasets with missing values, imputation can be performed using the k-Nearest Neighbour (`kNN`) algorithm. It is also possible to generate bootstrap samples from the raw dataset (Efron & Tibshirani, 1994).

Learning: structure learning can be performed using several state-of-the-art algorithms, namely the exact Silander-Myllymäki algorithm (SM, suitable for networks up to 20-30 nodes), the heuristic Max-min Hill-climbing (MMHC), and the Structural Expectation-Maximization algorithm (SEM). The scoring functions available are BDeu, AIC, BIC. Due to space constraints, we refer to (Koller & Friedman, 2009) for the relevant literature. In case of missing data, complete case analysis, available case analysis and learning after imputation are possible for SM and MMHC; SEM can learn a network directly from a dataset with missing values, iterating the imputation and structure learning steps until convergence. Structure learning can be helped by defining *layers* for the variables, preventing arcs from nodes in lower layers to nodes in upper layers, thus reducing the search space. Given the structure G, the set of parameters Θ can also be learnt, using Maximum-a-Posteriori (MAP) estimates.

Networks can be converted to *Partially Directed Acyclic Graphs* (PDAGs) representing their equivalence class. Bootstrap samples can be used to learn a *weighted* PDAG (WPDAG), where the weights reflect an estimate on the level of confidence of the edges.

Inference: for performing inference `bnstruct` provides the junction tree algorithm and the Expectation-Maximization algorithm (Koller & Friedman, 2009).

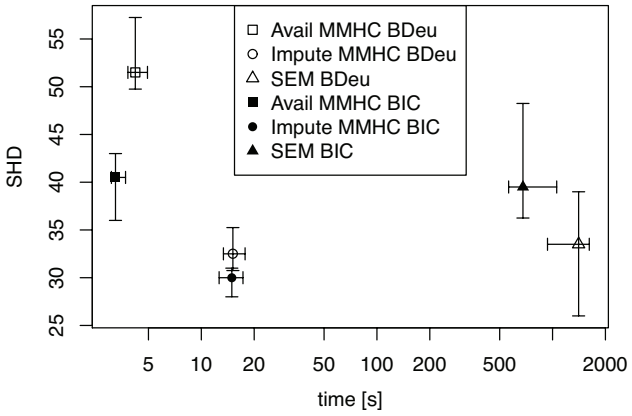


Figure 1. Structural Hamming Distance (Tsamardinos et al., 2006) vs. running time of i) available case analysis with MMHC, ii) kNN imputation followed by MMHC and iii) SEM, with both the BDeu and the BIC scoring functions, on 20 datasets with 1000 observations and 20% missingness sampled from the 37-nodes Alarm network (Beinlich et al., 1989).

Acknowledgments

This work has been partly funded by the MOSAIC European FP7 project (MOdels and Simulation techniques for discovering dIAbetes influence faCtors, Grant n. 600914); and the COMEX project within the Interuniversity Attraction Poles Programme of the Belgian Science Policy Office.

References

Beinlich, I. A., Suermondt, H. J., Chavez, R. M., & Cooper, G. F. (1989). *The alarm monitoring system: A case study with two probabilistic inference techniques for belief networks.* Springer.

Efron, B., & Tibshirani, R. J. (1994). *An introduction to the bootstrap.* CRC press.

Koller, D., & Friedman, N. (2009). *Probabilistic graphical models: principles and techniques.* MIT press.

McGeachie, M. J., Chang, H.-H., & Weiss, S. T. (2014). Cgbayesnets: conditional gaussian bayesian network learning and inference with mixed discrete and continuous data. *PLoS Comput Biol*, *10*.

Murphy, K. (2001). The bayes net toolbox for MATLAB. *Computing science and statistics*, *33*, 1024–1034.

Pearl, J. (1988). *Probabilistic reasoning in intelligent systems: networks of plausible inference.* Morgan Kaufmann.

Pérez-Bernabé, I., & Salmerón, A. (2015). *Motbfs: Learning hybrid bayesian networks using mixtures of truncated basis functions.* R package version 1.0.

Scutari, M. (2010). Learning Bayesian Networks with the bnlearn R Package. *Journal of Statistical Software*, *35*, 1–22.

Shah, A., & Woolf, P. (2009). Python environment for bayesian learning: inferring the structure of bayesian networks from knowledge and data. *JMLR*, *10*, 159–162.

Tsamardinos, I., et al. (2006). The Max-Min Hill-Climbing Bayesian network structure learning algorithm. *Machine learning*, *65*, 31–78.