Bioinformatics, 33(8), 2017, 1250–1252
doi: 10.1093/bioinformatics/btw807
Advance Access Publication Date: 21 December 2016
Applications Note

Systems biology

bnstruct: an R package for Bayesian Network structure learning in the presence of missing data

Alberto Franzin[1,2], Francesco Sambo[2,*] and Barbara Di Camillo[2]

[1]IRIDIA-CoDE, Université Libre de Bruxelles, 1050 Brussels, Belgium and [2]Department of Information Engineering, University of Padova, 35131 Padova, Italy

*To whom correspondence should be addressed.

Associate Editor: Jonathan Wren

Received on June 27, 2016; revised on December 12, 2016; editorial decision on December 14, 2016; accepted on December 15, 2016

Abstract

Motivation: A Bayesian Network is a probabilistic graphical model that encodes probabilistic dependencies between a set of random variables. We introduce bnstruct, an open source R package to (i) learn the structure and the parameters of a Bayesian Network from data in the presence of missing values and (ii) perform reasoning and inference on the learned Bayesian Networks. To the best of our knowledge, there is no other open source software that provides methods for all of these tasks, particularly the manipulation of missing data, which is a common situation in practice.
Availability and Implementation: The software is implemented in R and C and is available on CRAN under a GPL licence.
Contact: francesco.sambo@unipd.it
Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

Increasing attention has recently been devoted, in the bioinformatics community, to Bayesian Networks (Koller and Friedman, 2009), which are probabilistic graphical models that encode in a graph-based form the joint probability distribution of a set of random variables. A Bayesian Network (BN) $B = (\mathbf{X}, G, \Theta)$ is fully specified by a set of random variables $\mathbf{X}$, a Directed Acyclic Graph (DAG) G representing the dependencies between the variables (i.e. the *structure* of the network) and the set of *parameters* Θ specifying the conditional probability distribution of each variable according to the structure G.

Given a dataset, consisting of several observations, or *cases*, for a set of variables, a common problem is to learn the most probable network that may have generated the dataset. This task can be divided in two steps: learning the structure G of a plausible network B and learning the parameters Θ of the network G, from the same set of data.

In biological contexts, the problem is often complicated by missing values in the data due to out-of-threshold measurements, lost observations or impossibility of taking measures. To cope with this, several techniques can be employed: considering only the cases without any missing value (*complete case* analysis); using all cases without missing values for the variables under analysis (*available case* analysis); guessing missing values from the available data (*imputation*); iteratively applying missing values imputation and structure learning until convergence, as in the Structural Expectation Maximisation algorithm (SEM, Friedman, 1998).

We present an R package, bnstruct, that performs structure and parameter learning even in the presence of missing values using state-of-art algorithms for network learning and provides also methods for imputation, bootstrap re-sampling of the data and inference. bnstruct can handle both discrete and continuous variables in the dataset manipulation and imputation. However, as a design choice, learning is implemented with discrete variables alone, i.e. continuous variables are quantized after imputation.

As far as we know, there are no open source packages that use state-of-art algorithms for structure learning and inference in case of missing data: the bnlearn (Scutari, 2010), deal (Bottcher and Dethlefsen, 2013), CGBayesNets (McGeachie *et al.*, 2014), BNT (Murphy, 2001) and MoTBFs (Pérez-Bernabé and Salmerón, 2015)

packages perform structure learning but do not support missing data. PEBL (Shah and Woolf, 2009) cannot perform inference. SMILE (Druzdzel, 1999) and LibB (Friedman, 1998) are closed source.

2 Methods

Data imputation and bootstrap. In order to learn a BN, bnstruct requires both a dataset and some related metadata, such as name, cardinality and discreteness of each variable. In case the dataset contains some missing values, it is possible to extract a complete case dataset or to perform imputation with the k-Nearest Neighbour (kNN) algorithm. Bootstrap samples (Friedman *et al.*, 1999) can also be generated from the raw dataset.

Network learning. We provide bnstruct with state-of-the-art algorithms for structure learning, such as the Silander-Myllymäki complete algorithm (SM, Silander and Myllymäki, 2006), the constraint-based Max-Min Parent-and-Children algorithm (MMPC), the Hill-Climbing local search (HC), the Max-min Hill-climbing heuristic search algorithm (MMHC, Tsamardinos *et al.*, 2006) and the SEM algorithm (Friedman, 1997). SEM directly processes data with missing values, whereas the other algorithms perform structural learning after imputation with kNN. Scoring functions include the Bayesian Dirichlet equivalent uniform function (BDeu), the Akaike Information Criterion (AIC) and the Bayesian Information Criterion (BIC) (Koller and Friedman, 2009). Network DAGs can be converted to Partially Directed Acyclic Graphs (PDAGs) representing their corresponding equivalence classes (Koller and Friedman, 2009).

Constraints can be applied to the network structure G based on the problem knowledge: variables can be divided into *layers* and edges from lower to higher layers can be prevented: this is useful to learn Dynamic BNs. Once the structure is learned, distribution parameters Θ are computed as Maximum-a-Posteriori estimates. A network can aso be plotted or exported for external tools such as Cytoscape (Shannon *et al.*, 2003).

Bootstrap re-sampling can also be used to estimate a level of confidence on the learned edges (Friedman *et al.*, 1999). In this case, one network can be learned for each bootstrap sample and the resulting PDAGs can be aggregated in a *weighted PDAG* (WPDAG), where the confidence on each edge is estimated as the fraction of bootstrap samples from which the edge can be learned.

Inference. Bnstruct can also infer the estimated probability distribution of some variables, given evidence on the values of other observed variables. In this case, a junction tree (Koller and Friedman, 2009) is used. The Expectation-Maximization algorithm (Dempster *et al.*, 1977) is also implemented, which exploits a BN structure to iteratively estimate conditional probabilities from a dataset with missing values and impute missing values.

3 Experimental results

As an example, we considered the data from the screen visit of the Diabetes Control and Complications Trial (DCCT, The DCCT research group, 1993), where 1441 subjects suffering from Type 1 Diabetes (T1D) where screened for several anthropometric and metabolic risk factors for T1D complications. To search for probabilistic relations between the risk factors, we first divided them into three layers, based on domain information, and then learned 100 networks from 100 corresponding bootstrap samples of the data with the MMHC algorithm.

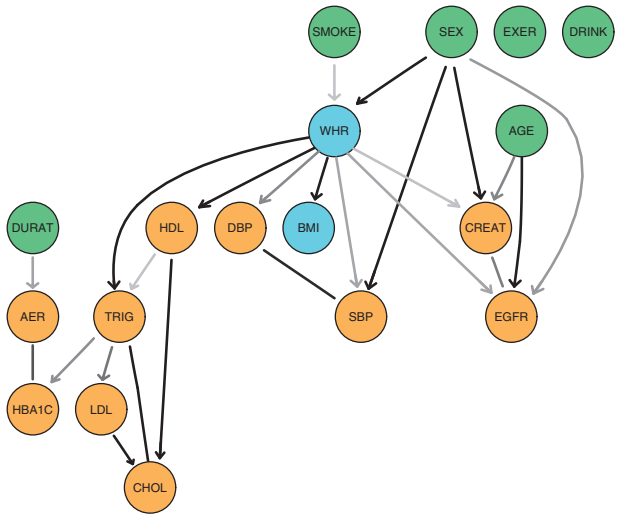

Fig. 1. WPDAG learned from the DCCT data. Nodes are divided into layers. Layer 1: smoke (SMOKE), gender (SEX), physical exercise (EXER), drinking (DRINK), age (AGE), T1D duration (DURAT). Layer 2: waist-hip ratio (WHR) and body mass index (BMI). Layer 3: systolic and diastolic blood pressure (SBP, DBP), HDL, LDL and total cholesterol (HDL, LDL, CHOL), triglycerides (TRIG), glycated hemoglobin (HBA1C), albumin excretion rate (AER), estimated GFR (EGFR) and creatinine (CREAT)

Figure 1 reports the final WPDAG, where the grey level of the edges is proportional to the estimated confidence on the probabilistic dependences and nodes are divided in the three layers. Several known dependencies are evident from the data, like the relation between WHR and the indices of renal complications (CREAT and EGFR) or between triglycerides and diabetes progression (HBA1C).

4 Conclusion

BNs learning plays a central role in bioinformatics and missing values are ubiquitous in bio-medical datasets. The R package bnstruct provides easy-to-use state-of-art algorithms to learn and reason with BNs from data that may contain missing values.

Funding

The 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.

Conflict of Interest: none declared.

References

Bottcher,S.G. and Dethlefsen,C. (2013) Learning Bayesian Networks with R. *In: Proceedings of the 3rd International Workshop on Distributed Statistical Computing*, 20–22.

Dempster,A.P. *et al.* (1977) Maximum likelihood from incomplete data via the EM algorithm. *J. R. Stat. Soc. Ser. B (Methodological)*, **39**, 1–38.

Druzdzel,M.J. (1999) Smile: Structural modeling, inference, and learning engine and genie: a development environment for graphical decision-theoretic models. In: *Aaai/Iaai*, pp. 902–903.

Friedman,N. (1997) Learning belief networks in the presence of missing values and hidden variables. In: *ICML*, Vol. **97**, pp. 125–133.

Friedman,N. (1998) The Bayesian Structural EM algorithm. In: *UAI-98*, pp. 129–138. Morgan Kaufmann Publishers Inc.

Friedman,N. *et al*. (1999) Data analysis with Bayesian networks: a bootstrap approach. In: *UAI-99*, pp. 196–205. Morgan Kaufmann Publishers Inc.

Koller,D. and Friedman,N. (2009) *Probabilistic Graphical Models: Principles and Techniques*. Cambridge: MIT Press.

McGeachie,M.J. *et al*. (2014) Cgbayesnets: conditional Gaussian Bayesian network learning and inference with mixed discrete and continuous data. *PLoS Comput. Biol*., **10**, e1003676.

Murphy,K. (2001) The Bayes net toolbox for MATLAB. *Comput. Sci. Stat*., **33**, 1024–1034.

Pérez-Bernabé,I. and 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. *J. Stat. Softw*., **35**, 1–22.

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

Shannon,P. *et al*. (2003) Cytoscape: a software environment for integrated models of biomolecular interaction networks. *Genome Res*., **13**, 2498–2504.

Silander,T. and Myllymäki,P. (2006) A simple approach for finding the globally optimal Bayesian network structure. In: *UAI-06*, pp. 445–452.

The DCCT Research Group. (1993) The effect of intensive treatment of diabetes on the development and progression of long-term complications in insulin-dependent diabetes mellitus. *N. Engl. J. Med*., **329**, 977–986.

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