Bayesian network data imputation with application to survival tree analysis

Paola M.V. Rancoita[a,b,c,*], Marco Zaffalon[b], Emanuele Zucca[d], Francesco Bertoni[c,d], Cassio P. de Campos[e]

[a]*University Centre for Statistics in the Biomedical Sciences, Vita-Salute San Raffaele University, Milan, Italy*
[b]*Dalle Molle Institute for Artificial Intelligence, Manno, Switzerland*
[c]*Institute of Oncology Research, Bellinzona, Switzerland*
[d]*Oncology Institute of Southern Switzerland, Bellinzona, Switzerland*
[e]*Queen's University Belfast, School of Electronics, Electrical Engineering and Computer Science, Belfast, UK*

Abstract

Retrospective clinical datasets are often characterized by a relatively small sample size and many missing data. In this case, a common way for handling the missingness consists in discarding from the analysis patients with missing covariates, further reducing the sample size. Alternatively, if the mechanism that generated the missing allows, incomplete data can be imputed on the basis of the observed data, avoiding the reduction of the sample size and allowing methods to deal with complete data later on. Moreover, methodologies for data imputation might depend on the particular purpose and might achieve better results by considering specific characteristics of the domain. The problem of missing data treatment is studied in the context of survival tree analysis for the estimation of a prognostic patient stratification. Survival tree methods usually address this problem by using surrogate splits, that is, splitting rules that use other variables yielding similar results to the original ones. Instead, our methodology consists in modeling the dependencies among the clinical variables with a Bayesian network, which is then used to perform data imputation, thus allowing the survival tree to be applied on the completed dataset. The Bayesian network is directly learned from the incomplete data using a structural expectation-maximization (EM) procedure in which the maximization step is performed with an exact anytime method, so that the only source of approximation is due to the EM formulation itself. On both simulated and real data, our proposed methodology usually outperformed several existing methods for data imputation and the imputation so obtained improved the stratification estimated by the survival tree (especially with respect to using surrogate splits).

Keywords: Bayesian networks, Data imputation, Missing data, Prognostic stratification, Survival tree.

1. Introduction

Retrospective clinical data are often used to identify features that can help in classifying patients into groups of similar survival and in predicting the survival outcome of patients (i.e. a prognostic patient stratification). The identification of classes of patients with a different clinical course or response to a specific treatment allows the design of the most appropriate approach for the management of each individual patient. The survival tree is a state-of-the-art method to stratify patients for predicting survival on the basis of available clinical parameters [3]. Although several algorithms exist for its estimation [7, 20, 21, 33, 18, 15, 11], the procedure always consists in finding, at each step, the best clinical variable able to divide the patients (with respect to the survival), so that the final stratification of the patients assumes a tree-like structure.

In retrospective studies, clinical and survival data may contain many missing values for several reasons. If the study includes data over a long period, some clinical parameters might not have been measured for

---

*Correspondence to: CUSSB, Vita-Salute San Raffaele University, Via Olgettina 58, 20132 Milan, Italy. Tel.: +390226433844.
Software freely available at: http://code.google.com/p/csda-dataimputation/
Email address: `rancoita.paolamaria@unisr.it` (Paola M.V. Rancoita)

some patients, because they were not systematically collected at diagnosis, and data might be missing in individual patients due to technical issues. More importantly, data might be missing in some particular subset of patients, causing biases in the analysis: for example, patients with a very aggressive course might have died before performing a test, or a test might have been skipped in patients expected to have a very good clinical course. Therefore, a retrospective study usually contains many missing covariate data, and this can heavily affect the statistical analysis, especially when the sample size is small. This issue worsens if the dataset contains many censored survival data. In fact, the missingness of the covariates added to the censoring issue increases the hardness of identifying an accurate prognostic stratification of the patients. In this work, we denote by missing data only the incomplete information happening in clinical and biological variables that are available in the analysis, and not the incomplete lifetime information of patients (censoring). A naive, still very used, approach to handle this issue consists in discarding all patients with missing variables from the analysis, decreasing the power of any model, which is clearly undesirable. Instead, survival tree procedures decide the best splits to define the tree using only the observed data for each variable, and they resort to surrogate splitting in case of missing values, that is, they use a splitting rule based on another variable which most resembles the behavior of the original missing one [1].

Another widely used approach to handle missing data is to impute the missing values, thus considering a complete dataset in further analyses [22]. The data imputation problem regards completing the dataset in some particular manner such that the important characteristics of the dataset are preserved. This is mostly done by assuming that missing data are *missing completely at random* (that is, their missingness is independent of both unobserved and observed data), which implies that data imputation can be safely performed by analyzing each variable separately. Widely used methods, such as single expected value imputation and single mode imputation, are based on this assumption. However, missing data in clinical datasets can be more realistically considered as *missing at random* instead of missing completely at random (that is, their missingness, conditional on the observed data, is independent of the unobserved values). In fact, some of the examples discussed before in this introduction are missing at random, but not completely at random. In the literature, many statistical approaches that account for the dependencies among covariates have been used for data imputation. In case of categorical or discrete variables (which is often the case for clinical parameters), these methods are usually based on maximum likelihood (ML) estimation of the joint distribution of the covariates from the partially classified contingency table built using the observed data [22]. Unfortunately, they suffer from the small sample size and tend to overfit, even with a small number of covariates, because they consider all dependencies among all variables.

We propose to use a methodology based on Bayesian networks as a way to impute accurately the missing data and improve the quality of the inference, especially in the application to survival tree analysis. For this application, our imputation method is employed only for imputing the covariates (without any knowledge about the survival data) and the survival tree is applied to the (supposedly accurate) completed dataset so obtained. A Bayesian network is a probabilistic graphical model that relies on a directed acyclic graph to encode the structured dependency among random variables and compactly represent a joint probability distribution. Learning and inference in these models benefit from fast and accurate procedures [19]. More specifically, learning a Bayesian network from data consists in searching for the structure of the network, as well as its parameters, such that some criterion of quality is maximized. The most common criterion for this purpose is the Bayesian Dirichlet Equivalent Uniform [14], which is based on maximizing the posterior probability of the structure given the data. Although this is a particularly challenging problem when data are incomplete, suitable algorithms do exist [12, 23, 36, 28, 26, 27]. These methods are mostly based on turning the incomplete data into a complete dataset (or even directly updating the sufficient statistics), and then recurring to particular methods for complete data. We adopt a meta-search composed of a few distinct methods [17, 9, 4, 34] that selects the best procedure to run depending on the number of covariates and running-time of the methods. The idea is to improve the score in the most efficient way, still with the guarantee of achieving optimality. After the Bayesian networks is learned, data imputation is performed by posterior expected means (and/or modes) computed from the estimated joint probability distribution (encoded by the Bayesian network). Standard belief updating (and/or maximum probable explanation) methods are used [19]. We use Bayesian networks in a similar way to previous studies [30, 10], but differently from them, we implement a global structure learning method that directly works with the expected sufficient

statistics instead of completing the data at each step of the structure learning process, thus achieving more accurate models. Our method for data imputation is freely available online at *http://code.google.com/p/csda-dataimputation/*.

The main contributions of this work are: (i) the design of a new framework that is able to capture the dependencies among variables, which is then used to estimate in a more accurate way the values of missing (categorical) covariates, and (ii) its comparison against other widely used methodologies to deal with missing data for survival tree analysis using both simulated data and real data of lymphoma patients. We empirically show that our proposed methodology may greatly improve: (i) the accuracy of data imputation (especially in case of strong dependencies among the variables) and (ii) the quality of results of the survival tree analysis, even in hard scenarios with a high percentage of missing and/or censoring data (which may occur in practice).

2. Bayesian networks for data imputation

In this section we formalize the problem of data imputation and describe how we apply Bayesian networks to perform such task, from the model learning to the missing values' inference. Since we aim at the application to clinical data and often clinical variables are binary, ordinal or discrete (with few values), we consider here only the case in which the covariates have a finite domain. In general, Bayesian networks can also be defined and learned for continuous variables (see, for example, Koller and Friedman [19]).

Let $\mathbf{X} = (X_1, \ldots, X_m)$ be a vector of categorical or discrete random variables assuming values in $\Omega_{\mathbf{X}} = \times_i \Omega_{X_i}$, where Ω_{X_i} is the state space of X_i, for every i, and $\Omega_{\mathbf{X}}$ is the cartesian product of them. Let $\mathbf{d} \in \Omega_{\mathbf{D}}$ be a realization (here, called also state configuration) of a subset of components $\mathbf{D}$ of $\mathbf{X}$. Suppose we have a dataset $\mathcal{D}$ with n samples, where, for each sample u, $\mathbf{D}_u$ represents the components of $\mathbf{X}$ which are observed in the sample, and $\mathbf{Z}_u$ the remaining missing ones. Thus, $\mathcal{D} = \{\mathbf{d}_1, \ldots, \mathbf{d}_n\}$ with $\mathbf{d}_u \in \Omega_{\mathbf{D}_u}$. We use $d_{u,i}$ to denote the observed value of the variable X_i in sample u, and we use $z_{u,i}$ to denote a completion of the missing value for X_i in the sample u, such that $(\mathbf{d}_u, \mathbf{z}_u) \in \Omega_{\mathbf{X}}$ is a complete configuration of $\mathbf{X}$, for some given value $\mathbf{z}_u$. We define $\mathcal{I}$ as the indicator function such that $\mathcal{I}_{\{condition\}} = 1$ if the *condition* is satisfied, and zero otherwise. For example, $\mathcal{I}_{\{x'=x\}} = 1$ if $x' = x$. As a particular case, we use $\mathcal{I}_{\{miss\ X_i;u\}}$ to indicate whether the variable X_i is missing in sample u. The data imputation problem regards completing each $\mathbf{d}_u$ with some $\mathbf{z}_u$ opportunely such that no missing value remains in the dataset. For instance, a single mean (and respectively a single mode) approach would take, for every (u, i):

$$z_{u,i} = \frac{\sum_u (1 - \mathcal{I}_{\{miss\ X_i;u\}}) \cdot d_{u,i}}{\sum_u (1 - \mathcal{I}_{\{miss\ X_i;u\}})} \quad \text{and} \quad z_{u,i} = \max_{x_i \in \Omega_{X_i}} \sum_u \mathcal{I}_{\{d_{u,i}=x_i\}}.$$

Note that single mode would keep the dataset discrete without changes in the variable's domains, while single mean would eventually introduce values that were not present before. The problem with these approaches is that they do not consider dependencies between the missing values and the observed ones. This is an acceptable behaviour if the missing values are assumed to be *missing completely at random* (or simply MCAR). However, in many cases one cannot be sure whether the missing data are MCAR, or they might even know that this is not the case. In such cases, the most usual assumption is that missing data are *missing at random* (or simply MAR), which means that although missing, the process which generated them may depend on the observed data (but not on the unobserved ones). Using the MAR assumption, single mode and single mean imputations are arguably too naive: no information from the other variables has been used in helping to estimate the missing values.

Given the MAR assumption, one can perform data imputation such that the likelihood of the observed data, or its posterior probability (as we do here), is maximized. First, they compute $\mathcal{M} = \text{argmax}_{\mathcal{M}'} \Pr_{\mathcal{M}'}(\mathcal{D}|\mathcal{M}')$, where $\mathcal{M}$ is the model which maximizes the probability of the data, and then they use the model $\mathcal{M}$ to compute the expected values (or the mode) for the imputation of the missing data. For every u, i:

$$z_{u,i} = \text{E}_{\mathcal{M}}[X_i|\mathbf{d_u}] \quad \text{or} \quad \mathbf{z}_u = \max_{\mathbf{z}_u \in \Omega_{\mathbf{Z}_u}} \Pr_{\mathcal{M}}(\mathbf{z}_u|\mathbf{d}_u),$$

3

where $\mathrm{E}_{\mathcal{M}}$ and $\mathrm{Pr}_{\mathcal{M}}$ are, respectively, the expectation and the probability with respect to $\mathcal{M}$. However, when the number of variables in $\mathbf{X}$ is large, any direct approach that must handle MAR data becomes intrinsically exponential, because even the specification of the joint probability distribution is prohibitive. This situation would happen, for instance, with a direct application of either expectation-maximization or data augmentation, as well as any other method that deals with a full specification of the joint distribution [22]. In such cases, one can resort to more compact representations of the joint distribution, such as the Bayesian networks, which can be learned from the data.

We point out that, in general, when the data imputation is based on the conditional distribution of one variable with respect to the others, often only some variables are considered in the conditioning. In that case, when the imputation is performed with the conditional expected value, the resulted imputed dataset may present altered associations among the variables, especially with those not considered in the conditional distribution. One way to overcome this issue would be to impute the data with a random draw from the conditional distribution, but the imputation with a single sampling would not reflect the sampling variability. Thus, usually multiple imputations are preferable, but they require a method for combining the inference across the imputed datasets [22]. Instead, in our methodology, we estimate the joint distribution among all variables, thus the associations among the variables should not be affected by such issue.

2.1. Bayesian network definition

A Bayesian network $\mathcal{M}$ is a probabilistic graphical model that relies on a structured dependency among random variables to represent a joint probability distribution in a compact and efficient manner. It represents a joint probability distribution $\mathrm{Pr}_{\mathcal{M}}$ over a collection of discrete random variables $\mathbf{X}$. It can be defined as a triple $\mathcal{M} = (\mathcal{G}, \mathbf{X}, \mathcal{P})$, where $\mathcal{G} = (V_{\mathcal{G}}, E_{\mathcal{G}})$ is a directed acyclic graph (DAG) with $V_{\mathcal{G}}$ a collection of m nodes associated to the random variables $\mathbf{X}$ (a node per variable), and $E_{\mathcal{G}}$ a collection of arcs; $\mathcal{P}$ is a collection of conditional probabilities $\mathrm{Pr}_{\mathcal{M}}(X_i|PA_i)$ where PA_i denotes the parents of X_i in the graph (PA_i may be empty), corresponding to the relations of $E_{\mathcal{G}}$. In a Bayesian network, the Markov condition states that every variable is conditionally independent of its non-descendants given its parents. This structure induces a joint probability distribution by the expression $\mathrm{Pr}_{\mathcal{M}}(X_1, \ldots, X_m) = \prod_i \mathrm{Pr}_{\mathcal{M}}(X_i|PA_i)$. We define $r_i \geq 2$ as the number of values in Ω_{X_i}, r_{PA_i} as the number of possible configuration of the parent set, that is, $r_{PA_i} = \prod_{X_l \in PA_i} r_l$, and $\boldsymbol{\theta}$ as the entire vector of parameters such that $\theta_{ijk} = \mathrm{Pr}_{\mathcal{M}}(X_i = x_{i,k}|PA_i = pa_{i,j})$, where $x_{i,k} \in \Omega_{X_i}$ and $pa_{i,j} \in \Omega_{PA_i}$, for $i \in \{1, \ldots, m\}$, $k \in \{1, ..., r_i\}$, $j \in \{1, ..., r_{PA_i}\}$.

Given a dataset $\mathcal{D}$ with n samples, the structure learning problem in Bayesian networks is to find a DAG $\mathcal{G}$ that maximizes a given score function, that is, we look for $\mathcal{G}^* = \mathrm{argmax}_{\mathcal{G} \in \boldsymbol{\mathcal{G}}} s_{\mathcal{D}}(\mathcal{G})$, with $\boldsymbol{\mathcal{G}}$ the set of all DAGs over node set $\mathbf{X}$. In this paper, we consider the score function $s_{\mathcal{D}}$ to be the Bayesian Dirichlet Equivalent Uniform (BDeu) criterion [2, 4]. As done before in the literature, we assume parameter independence and modularity [14]. The idea is to compute a score based on the posterior probability of the structure $\mathrm{Pr}(\mathcal{G}|D)$. For that purpose, the following score function is used:

$$s_{\mathcal{D}}(\mathcal{G}) = \log \left(p(\mathcal{G}) \cdot \int p(\mathcal{D}|\mathcal{G}, \boldsymbol{\theta}) \cdot p(\boldsymbol{\theta}|\mathcal{G}) d\boldsymbol{\theta} \right),$$

where the logarithmic is often used to simplify computations, $p(\boldsymbol{\theta}|\mathcal{G})$ is the prior of $\boldsymbol{\theta}$ for a given graph $\mathcal{G}$, assumed to be a Dirichlet with hyper-parameters $\boldsymbol{\alpha} = \{\alpha_{ijk}\}_{ijk}$:

$$p(\boldsymbol{\theta}|\mathcal{G}) = \prod_{i=1}^{m} \prod_{j=1}^{r_{PA_i}} \Gamma(\alpha_{ij\bullet}) \prod_{k=1}^{r_i} \frac{\theta_{ijk}^{\alpha_{ijk}-1}}{\Gamma(\alpha_{ijk})},$$

where $\alpha_{ijk} = \frac{\alpha^*}{r_{PA_i} r_i}$, for all j, k, $\alpha_{ij\bullet} = \frac{\alpha^*}{r_{PA_i}}$, for all j, and α^* is the only free hyper-parameter, usually referred to as the Equivalent Sample Size (ESS). We assume that there is no preference for any graph, so $p(\mathcal{G})$ is uniform and vanishes in the computations. Under these assumptions and if the data are complete, it has been shown [4] that,

$$s_D(\mathcal{G}) = \log \prod_{i=1}^{m} \prod_{j=1}^{r_{PA_i}} \frac{\Gamma(\alpha_{ij\bullet})}{\Gamma(\alpha_{ij\bullet} + N_{ij})} \prod_{k=1}^{r_i} \frac{\Gamma(\alpha_{ijk} + N_{ijk})}{\Gamma(\alpha_{ijk})}, \tag{1}$$

4

where N_{ijk} indicates how many elements of $\mathcal{D}$ contain both $X_i = x_{i,k}$ and $PA_i = pa_{i,j}$. The values $\{N_{ijk}\}_{ijk}$ depend on the graph $\mathcal{G}$ (more specifically, they depend on the parent set PA_i of each X_i), so a more precise notation would be to use $N_{ijk}^{PA_i}$ instead of N_{ijk}, which we avoid to make things simpler. Moreover, note that we do not really need to know each element of $\mathcal{D}$ to compute the score function, but it is sufficient to have $\boldsymbol{N} = \{N_{ijk}\}_{ijk}$.

2.2. Learning Bayesian networks

Learning the structure of these networks from data is a very challenging problem, especially when data are incomplete. Assuming that missing values are MAR, we could marginalize missing variables to obtain a function of the observed ones. However, this function becomes hard to evaluate, and other problems arise: for instance, methods that build a cache of local scores based on score decomposability [8] cannot be applied anymore. Another solution is the expectation-maximization (EM) algorithm, which was extended to work on the structure learning problem [12]. The idea is to use the E-step to compute the expected counts (which is an *expected* sufficient statistics for the data) given the current structure and parameters of the network, and then apply such expected counts to learn a new structure. This is done iteratively until no improvement in the structure or parameters is possible. A possible version of the algorithm is as follows:

Structure learning algorithm
Input: dataset $\mathcal{D}$
Output: Bayesian network $\mathcal{M}$

1. Choose an initial guess $\mathcal{G}^0$ and parameters $\boldsymbol{\theta}^0$.
2. For $t = 0, 1, \ldots$
    (a) Set the structure of $\mathcal{M}^t$ to be $\mathcal{G}^t$ and run a parameter learning method for incomplete data $\mathcal{D}$ to find new parameters $\boldsymbol{\theta}$ for $\mathcal{M}^t$, using its structure and the parameters $\boldsymbol{\theta}^t$.
    (b) Compute $\boldsymbol{N}$ using $\mathcal{M}^t$ to deal with the missing values such that each missing $z_{u,i}$ is associated to the probability mass function $\Pr_{\mathcal{M}^t}(X_i | \mathbf{d}_u)$. Set $s' = s_N(\mathcal{G}^t)$.
    (c) Run the learning method that searches for a $\mathcal{G}^{t+1}$ that maximizes s_N. In fact, we stop this step as soon as the value of s_N becomes greater than s'. If there is no $\mathcal{G}^{t+1}$ that improves the previous score s', go to step 3.
3. Set $\mathcal{M} = \mathcal{M}^t$.

In order to optimize the score function over the space of possible graphs $\boldsymbol{\mathcal{G}}$ in the step (2c) of the algorithm, we have implemented a few algorithms available in the literature, and a meta-algorithm that is able to select the best option according to some problem characteristics and/or running time limits. Namely, we have implemented the K2 local search of Cooper and Herskovits [4], the branch-and-bound (BB) procedure in de Campos and Ji [9], a pruned dynamic programming (DP) based on Silander and Myllymaki [34], and the linear integer programming (IP) of Jaakkola et al. [17]. These methods all require the dataset to be complete, or at least to have a way to compute the sufficient statistics of $\mathcal{D}$ in the step (2b) of the algorithm. We associate to each missing value $z_{u,i}$ the corresponding distribution $p(Z_{u,i})$, using $p(Z_{u,i}) = \Pr_{\mathcal{M}}(X_i | \mathbf{d}_u)$, just as if the observation for $z_{u,i}$ had been split into its states proportionally to $\Pr_{\mathcal{M}}(X_i | \mathbf{d}_u)$. Besides that, we employ the properties of the BDeu score of de Campos and Ji [8] in order to reduce the search space even before calling the structure learning methods.

Past literature experiments [6, 9, 17] indicate that DP is the fastest method for small values of m (fewer than $15 - 20$ variables), IP is the best method from $15 - 20$ to a hundred, and IP and BB are anytime algorithms, so can be run even with large datasets, and then the accuracy keeps improving with time. K2 is the only non-exact method (that is, not guaranteed to converge to a global maximum solution), but it is very efficient, so we try to use the K2 search as much as possible, that is, we use K2 as long as it can find an improving solution with respect to the previous run (if it cannot, then a globally optimal method is used). Therefore, our structure learning starts by running K2 with random initial guesses until a time limit or convergence of K2 is reached. If an improving solution is found, the method stops and returns it. If K2 fails to find an improvement, one of DP, IP or BB is run, depending on the number of variables. DP and IP are given a certain time-limit, which if surpassed, BB is also called.

Because the structure learning finds an improving solution (or the computation finishes), the method increases the score at each iteration, and so it converges. Until recently, exact methods to find the best structure were simply prohibitive unless for toy examples, hence structural EM had been developed using only a local search on the space of structures, that is, step (2c) had never been performed with a global search method. As described, we can also make use of approximate methods such as the K2, but we will never decide to stop the search prematurely if there is an improving solution, because in such case we would have run also a global method, which would eventually decide for sure whether there is in fact an improving solution or not. Hence, the source of approximation in this enhanced structural EM is due solely to the use of the EM, which is intrinsic of such idea.

2.3. Completing the data

In order to perform data imputation, we use the Bayesian network learned by the procedure described in Section 2.2. After all, the Bayesian network we learn simply represents a joint distribution for the variables with the compromise between fitting the (incomplete) data $\mathcal{D}$ and keeping the description compact (thus efficient to compute with). Data imputation can be naturally performed by using either the posterior expected mean (for discrete or ordinal variables):

$$z_{u,i} = E[X_i|\mathbf{d}_u] = \sum_{x_i} x_i \mathrm{Pr}_{\mathcal{M}}(x_i|\mathbf{d}_u), \tag{2}$$

which corresponds to the *belief updating* query in the learned Bayesian network [19], or the posterior expected mode (for either discrete or categorical variables):

$$\boldsymbol{z}_u = \max_{\boldsymbol{x} \in \Omega_{Z_u}} \mathrm{Pr}_{\mathcal{M}}(\boldsymbol{x}|\mathbf{d}_u),$$

which corresponds to the *most probable explanation* query in the Bayesian network. In both cases, the learned structure is used to compute the desired values in an efficient way (more precisely, their complexity depends exponentially on the treewidth of the network, a measure of the similarity of the network's graph to a tree). In the particular case of missing values for binary variables, the data completion by the expected value in Equation (2) can be used even if the variable is nominal, because it represents the preference between categories (in terms of frequencies of choosing the category one).

We end this section by discussing on other methods to perform data imputation. Di Zio et al. [10] also use Bayesian networks for data imputation, but they force the network (and the imputation procedure) to follow a pre-defined order among the variables in the domain. This constraint is known to restrict the learning to sub-optimal Bayesian networks. Romero and Salmerón [30] use an imputation method that resembles the learning procedure of Section 2.2. The main difference is their use of k-best most probable explanation sequences [24] (for each sample in the dataset) to fill in the missing values at each iteration (they sample an instantiation from the k best according to their probabilities, which is done to mimic a joint data augmentation approach, while keeping the computational complexity low), instead of working with the (expected) sufficient statistics computed from the data. Riggelsen and Feelders [28] propose ideas to learn Bayesian networks using imputation, but adopt a data augmentation procedure [37] where the missing data are imputed with an importance sampling approach. These ideas seem well suitable when the number of samples are over a thousand [27], which is rarely the case of most medical datasets. Finally, to the best of our knowledge, all previous attempts have used local search methods to deal with the structure learning problem [10, 30, 26, 28, 27], while we use a globally optimal procedure.

3. Survival tree for prognostic patient stratification

Let us consider the right-censored survival data of n patients. The observed time of patient u is denoted by $Y_u = \min\{T_u, C_u\}$, where T_u is the failure time (or event time) and C_u is the censoring time of that patient. The censoring indicator is denoted by $\delta_u = \mathcal{I}_{\{T_u \leq C_u\}}$, that is, $\delta_u = 1$ when the event time is

observed (i.e. $Y_u = T_u$). Hence, for each patient u in the dataset, his/her survival information is represented by the pair (Y_u, δ_u). As usual, we assume censoring to be independent of the observations.

In survival analysis, it is of interest to estimate the survival function $S(t) = P(T > t)$, for all $t \geq 0$, which represents the probability to be event-free up to time t. More in general, this function is studied conditional on the values of some covariates $\mathbf{X} = (X_1, \ldots, X_m)$, that is $S(\cdot|\mathbf{X})$, and in our context the covariates may represent clinical and/or biological parameters known at diagnosis. For the definition of S, it is common to use a proportional hazards model:

$$
\begin{aligned}
S(t|\mathbf{X}) &= e^{-\int_0^t \lambda(s|\mathbf{X})ds} \\
&= e^{-h(\mathbf{X})\int_0^t \lambda_0(s)ds}, \qquad t \geq 0,
\end{aligned} \tag{3}
$$

by assuming that $\lambda(s|X_1, \ldots, X_m) = \lambda_0(s)h(\mathbf{X})$ (i.e. $h(\mathbf{X}) = \lambda(s|\mathbf{X})/\lambda_0(s)$), for all $s \geq 0$, where λ and λ_0 are the hazard and the baseline hazard function, respectively. As example, choosing $h(\mathbf{X})$ to be $\exp(\beta_1 X_1 + \ldots + \beta_m X_m)$ gives us the Cox proportional hazards model [5].

3.1. Survival tree

In clinical studies, it is often more interesting to estimate the survival function depending only on few subsets of the space defined by the covariates (that is, few combinations of sets of values or intervals of the covariates), instead of a full continuous model, such as the Cox proportional hazards one. In this way, the patients can be divided in fewer subgroups based on the covariates (that is, each subset in the covariate space correspond to a subgroup) and a more accurate estimation of S can be done for each subgroup. Consequently, we aim at defining a set of groups of patients $\mathbb{G} = \{g_1, \ldots, g_{n_G}\}$, based on the values of $\mathbf{X} = (X_1, \ldots, X_m)$, and at estimating $S(.|G = g)$, $\forall g \in \mathbb{G}$, where G is the variable representing the group to which patients belong. By creating more homogeneous groups of patients, we obtain a better survival prediction of a new patient, given its covariates. This problem can be called *prognostic patient stratification*. For instance, such analysis allows the identification of patients with predicted poor survival, which might therefore take advantage from experimental therapies or more intensive regimens.

A well-known and state-of-the-art method for the prognostic patient stratification is the survival tree, which is a regression tree with a split-function that is able to deal with censored data. In this contest, the groups are defined by combinations of the values of the covariates defined by the estimated tree-like structure. The main advantages of such method are that: 1) it selects the covariates useful in the model, considering their dependency; 2) it automatically defines the best cut-points for splitting the values of each covariate in sets or intervals used to form the definition of the groups; 3) it gives a highly informative output since the tree structure also shows a hierarchical dependency among the selected variables [3]. These characteristics make it more appealing and powerful than more standard procedures, like stepwise Cox's regression and elastic net Cox's regression [35].

Several algorithms exist for the estimation of the survival tree [7, 20, 21, 33, 18]. Most of them are essentially based on the CART algorithm [1], but differ in the definition of the split function. Three well-known ones are based on: the one-step full exponential likelihood deviance [20], also implemented in the widely used R package `rpart`; the two-sample log-rank statistics [33]; and the least-squares with martingale residuals [18]. In the literature it has not been proven any strong outperformance of one of them over the others [20, 18]. Another regression tree methodology is the conditional inference tree [15], which is implemented in the R package `party`. In case of a censored response variable, the procedure is based on conditional log-rank tests [25] with multiplicity adjusted p-values. Although this method has good theoretical properties, it has been shown that its performance is comparable with that of the algorithm implemented in `rpart`, in case of categorical and numerical responses [15, 32]. As far as we know, these algorithms have not been fully compared in survival settings where the presence of a high percentage of censored data may increase the difficulty of the estimation. Since the comparison of different survival tree methodologies is out of the aim of this article and none has been shown to always outperform the others, here we employ the procedure implemented in `rpart`, which is the most widely used.

4. Results

In order to evaluate the performance of the Bayesian network for data imputation, we compare it with other existing methods cited in Sections 1 and 2, using simulated data with characteristics similar to real clinical data. Overall, the methodologies that we consider are:

- the imputation with the expected mean (`indep_E`) or the mode (`indep_M`), by estimating them, separately for each covariate, using the corresponding available training data only for that given covariate;

- the imputation with the expected mean (`FL_E`) or the mode (`FL_M`), computed on the basis of the ML estimate of the joint distribution of the covariates from the partially classified contingency table [22] (R package `cat`);

- the imputation with the method of Romero and Salmerón [30], described in Section 2.3 (`impute_BN`);

- the imputation with the posterior expected mean (`BN_E`) or the posterior mode (`BN_M`), computed on the basis of the posterior joint distribution of the covariates estimated by our learned Bayesian network, which is the methodology we propose in Section 2 (the method is freely available online at this address: *http://code.google.com/p/csda-dataimputation/*).

Since our main goal is the application of data imputation to clinical studies, further to the evaluation of the accuracy in the imputed data themselves, we use the simulated data also to assess whether data imputation is able to improve the prognostic patient stratification obtained by the survival tree. Namely, we compare the accuracy of the survival prediction between the model found by the survival tree using the surrogate splits (`SS`) for dealing with missing values (which is the common procedure in the regression tree algorithms, as explained in Section 1), and the model estimated by the survival tree when applied to the dataset after the imputation. As explained in Section 3, we use the survival tree algorithm implemented in the R package `rpart` [18].

Finally, we show the performance of our proposed methodology in the application to real datasets of two diseases (diffuse large B-cell lymphoma and marginal zone lymphoma). For both of them, a separate test dataset was available for the assessment of the results, as it will be detailed later on.

4.1. Simulation study

We have used the simulated data with two purposes: 1) to compare our method (based on Bayesian networks) and others in the literature with respect to the accuracy in the data imputation; 2) to show that the imputation can improve the final model estimated by the survival tree for patient prognostic stratification. Since often clinical parameters are represented by binary variables (or, sometimes, by ordinal or discrete variables with few values) and our focus is on the application to clinical data, our simulated datasets consist of ten binary covariates $(X_1, \ldots, X_{10})$ and the survival variables Y and δ (following the notation defined in Section 3). We define the following exponential model for describing the relationship between the covariates and the survival function:

$$S(t|\mathbf{X}) = e^{-h(\mathbf{X})t}, \qquad \text{for all } t \geq 0, \tag{4}$$

where

$$h(\mathbf{X}) = \begin{cases} h_1 = 0.25, & \text{if } (X_1 = 0, X_2 = 0) \\ h_2 = 1, & \text{if } (X_1 = 0, X_2 = 1, X_4 = 0, X_6 = 0) \\ h_3 = 1.5, & \text{if } (X_1 = 0, X_2 = 1, X_4 = 1) \\ h_4 = 2.25, & \text{if } (X_1 = 1) \\ h_5 = 2.5, & \text{if } (X_1 = 0, X_2 = 1, X_4 = 0, X_6 = 1), \end{cases} \tag{5}$$

that is, $S(t|G = g) = e^{-h_g t}$, for all $t \geq 0$ and $g = 1, \ldots, 5$. According to the model defined through Equations (4) and (5), the patients are divided into five groups, on the basis of only four out of the ten covariates, and each group corresponds to a different $h(X_1, \ldots, X_{10})$ in the proportional hazards model of Equation (3). Moreover, it can be easily seen that the model for the definition of the groups can be
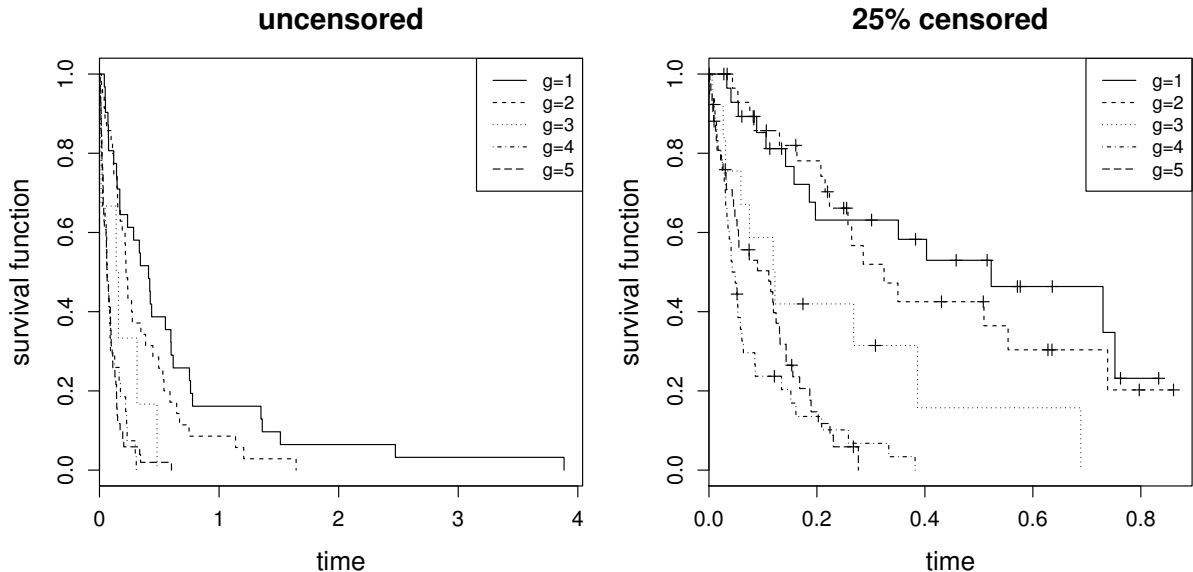
Figure 1: Examples of Kaplan-Meier estimated survival functions of the five groups in the simulated data. The estimation was done in two training datasets of model B, with the groups defined as in Equations 4 and 5, and 0% or 25% censored data, respectively.

represented by a tree-like structure, given the particular way in which we define the conditions in the equation. An example of Kaplan-Meier estimation of the survival function for the five groups is given in Figure 1. As commonly assumed in the literature (e.g. Hothorn et al. [16]), the model is defined with a censoring time C uniformly distributed on $[0, \gamma]$, and then $Y = \min(T, C)$ and $\delta = \mathcal{I}_{\{T \leq C\}}$. The value of γ is chosen depending on the distribution model of the covariates (explained below) in order to obtain the desired percentage of censored data (see Table 1). We allow the training datasets to have 0%, 25% or 50% of censored patients, while the patients in the test datasets are all uncensored, which implies a better evaluation of the survival prediction.

Similarly to real clinical data, the covariates are not all independent of each other and Figure 2 shows the relationships among them. For the generation of the data, we used two specifications of conditional probabilities (here called *dependency models*) that define the quantitative part of the Bayesian network of Figure 2: the first one represents weak dependencies (called *model A*) and the other one strong dependencies (called *model B*). The strength of the dependencies is accounted by how the conditional probability values are defined (see Table 1).

Regarding the generation of the missing data, we use a MAR model (applied to the complete dataset), which allows missing values only in five out of the ten covariates: X_1, X_2, X_6, X_8 and X_{10}. The probability of having a missing value in the variables X_1, X_2, X_8 and X_{10} is defined conditionally on the observed value in X_3, X_4, X_9 and X_7, respectively (see Table 2). These added relationships among the variables increase the difficulty of the data imputation task. The set of probabilities for the MAR model are defined for each dependency model, in order to obtain 15%, 25%, or 40% of missing values in each of the five covariates X_1, X_2, X_6, X_8 and X_{10}.

In order to resemble even more the difficulties usually encountered in the analysis of real clinical datasets, the sample size is set to only 150 in each training dataset, but is set to 200 in each test dataset to obtain a reasonably accurate evaluation of the performance. As for the generation of the data, we first simulated 100 training datasets of the covariates for each dependency model (models A and B). Then we applied (to each complete training dataset) the suitable MAR models to derive other three datasets corresponding to 15, 25 and 40 percentage of missing values. The simulated survival data are obtained by applying the survival
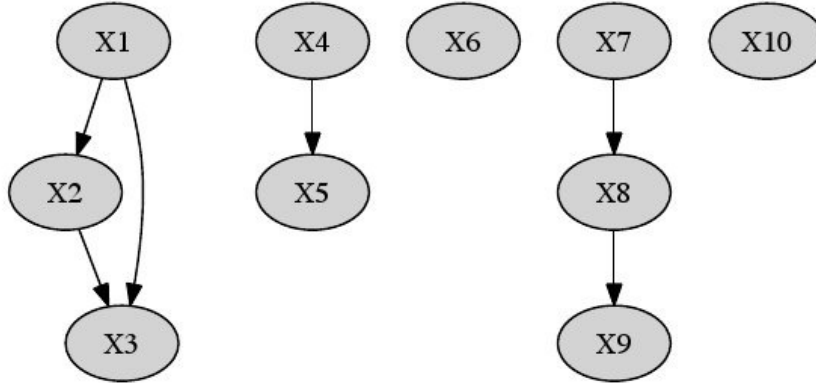
9

Figure 2: Dependency structure of the ten binary covariates in the simulated data.

Table 1: Definition of the two dependency models in the simulation data, along with the definition of their γ parameter in the distribution of the two censoring times (corresponding to have 25% and 50% of censoring data). $\mathcal{B}$ and $\mathcal{U}$ denote the Bernoulli and the uniform distributions, respectively.

	model A	model B
variable	**distributions in the dependency model**	
X_1	$\mathcal{B}(0.20)$	$\mathcal{B}(0.20)$
$X_2\|X_1 = 0$	$\mathcal{B}(0.75)$	$\mathcal{B}(0.75)$
$X_2\|X_1 = 1$	$\mathcal{B}(0.26)$	$\mathcal{B}(0.95)$
$X_3\|(X_1 = 0, X_2 = 0)$	$\mathcal{B}(0.37)$	$\mathcal{B}(0.77)$
$X_3\|(X_1 = 1, X_2 = 0)$	$\mathcal{B}(0.09)$	$\mathcal{B}(0.07)$
$X_3\|(X_1 = 0, X_2 = 1)$	$\mathcal{B}(0.47)$	$\mathcal{B}(0.77)$
$X_3\|(X_1 = 1, X_2 = 1)$	$\mathcal{B}(0.75)$	$\mathcal{B}(0.76)$
X_4	$\mathcal{B}(0.10)$	$\mathcal{B}(0.10)$
$X_5\|X_4 = 0$	$\mathcal{B}(0.14)$	$\mathcal{B}(0.05)$
$X_5\|X_4 = 1$	$\mathcal{B}(0.77)$	$\mathcal{B}(0.75)$
X_6	$\mathcal{B}(0.60)$	$\mathcal{B}(0.60)$
X_7	$\mathcal{B}(0.39)$	$\mathcal{B}(0.96)$
$X_8\|X_7 = 0$	$\mathcal{B}(0.25)$	$\mathcal{B}(0.03)$
$X_8\|X_7 = 1$	$\mathcal{B}(0.36)$	$\mathcal{B}(0.89)$
$X_9\|X_8 = 0$	$\mathcal{B}(0.43)$	$\mathcal{B}(0.90)$
$X_9\|X_8 = 1$	$\mathcal{B}(0.74)$	$\mathcal{B}(0.18)$
X_10	$\mathcal{B}(0.77)$	$\mathcal{B}(0.90)$
% censoring	**γ in the censoring distribution $\mathcal{U}[0, \gamma]$**	
25%	0.988	0.988
50%	0.299	0.299

Table 2: Distribution of the missing values in the variables X_1, X_2, X_6, X_8 and X_{10}, depending on the values of X_3, X_4, X_9 and X_7. Only in two cases we needed to slightly change the definition of the distribution in other to achieve the requested amount of missing data. p_{miss} denotes the percentage of missing values of the variable requested in the scenario.

variable	model of the distribution of the missing values	
X_6	$P(\mathcal{I}_{\{miss\ X_6\}}) = p_{miss}$	
X_1	$P(\mathcal{I}_{\{miss\ X_1\}}|X_3 = 1) = \frac{p_{miss}}{P(X_3=1)}$	$P(\mathcal{I}_{\{miss\ X_1\}}|X_3 = 0) = 0$
X_2	$P(\mathcal{I}_{\{miss\ X_2\}}|X_4 = 1) = \frac{p_{miss}}{P(X_4=1)}$	$P(\mathcal{I}_{\{miss\ X_2\}}|X_4 = 0) = 0$
X_8	$P(\mathcal{I}_{\{miss\ X_8\}}|X_9 = 1) = \frac{p_{miss}}{P(X_9=1)}$	$P(\mathcal{I}_{\{miss\ X_8\}}|X_9 = 0) = 0$
$X_8{}^{(a)}$	$P(\mathcal{I}_{\{miss\ X_8\}}|X_9 = 1) = \frac{p_{miss}}{4P(X_9=1)}$	$P(\mathcal{I}_{\{miss\ X_8\}}|X_9 = 0) = \frac{3p_{miss}}{4P(X_9=0)}$
X_{10}	$P(\mathcal{I}_{\{miss\ X_{10}\}}|X_7 = 1) = \frac{p_{miss}}{P(X_7=1)}$	$P(\mathcal{I}_{\{miss\ X_{10}\}}|X_7 = 0) = 0$
$X_{10}{}^{(b)}$	$P(\mathcal{I}_{\{miss\ X_{10}\}}|X_7 = 1) = \frac{p_{miss}}{4P(X_7=1)}$	$P(\mathcal{I}_{\{miss\ X_{10}\}}|X_7 = 0) = \frac{3p_{miss}}{4P(X_7=0)}$

[a] Distribution used only in model B with $p_{miss} = 0.40$.

[b] Distribution used only in model A with $p_{miss} = 0.40$.

model explained at the beginning of this section to the corresponding complete dataset, for each desired percentage of censored data (0%, 25%, 50%). For the evaluation of the improvement of the survival tree estimation brought by data imputation, we generated 100 complete test datasets by each dependency model (models A and B), with uncensored survival data. For the assessment of the performance of the methods in the data imputation only, from these complete datasets we generated the corresponding datasets with 15, 25 and 40 percentage of missing values, by applying the same MAR models as for the training datasets.

In order to compare the accuracy of the methods in the data imputation, firstly we applied them to the training datasets for all settings (that is, combinations of dependency model and percentage of missing data), in order to estimate the joint distribution of the covariates. Secondly, we imputed the missing data in the corresponding incomplete test dataset. We assessed the goodness of the estimated joint distributions by comparing the square of the Hellinger distance,

$$H^2(Pr, \widehat{Pr}) = \frac{1}{2} \sum_{\mathbf{x} \in \Omega_{\mathbf{X}}} \left(\sqrt{Pr(\mathbf{X} = \mathbf{x})} - \sqrt{\widehat{Pr}(\mathbf{X} = \mathbf{x})} \right)^2,$$

and the Kullback-Leibler divergence,

$$D(\widehat{Pr}\|Pr) = \sum_{\mathbf{x} \in \Omega_{\mathbf{X}}} Pr(\mathbf{X} = \mathbf{x}) \log \left(\frac{Pr(\mathbf{X} = \mathbf{x})}{\widehat{Pr}(\mathbf{X} = \mathbf{x})} \right),$$

obtained with the four methods we considered in this study (see Figure 3 and Supplementary Figure 1). The paired Wilcoxon signed-rank test was calculated for assessing whether our methodology achieved a better or similar accuracy to the other methods with respect to the joint distribution estimation (see Supplementary Tables 1 and 2). The test was always significant, except for model A with 25% and 40% of missing data. In these cases, indep performed better than ours. In fact, in presence of a high amount of missing data, learning an accurate dependency structure is very hard, also because of the small sample size of the training set. Moreover, in model A, we have weak dependencies among the variables. Thus, any reasonably sophisticated method (including ours) will hardly outperform a simpler method such as indep, since it cannot obtain enough information from the data to estimate a very accurate model of the structure. The accuracy of the joint distribution estimation reflects how close we got to the global information encoded by the data instead of looking to the particular pointwise estimation of the missing values. To also evaluate the accuracy of the estimation of each missing value, we employed the mean square error (MSE) between the imputed data in the incomplete test dataset and their true values in the corresponding complete test dataset (only considering the cells with missing values in the incomplete one and obviously the true values were not available to the imputation methods). Figure 4 shows the boxplots of the MSE for all methods in all scenarios. Moreover, we also computed the paired Wilcoxon signed-rank test to assess if BN_E performed
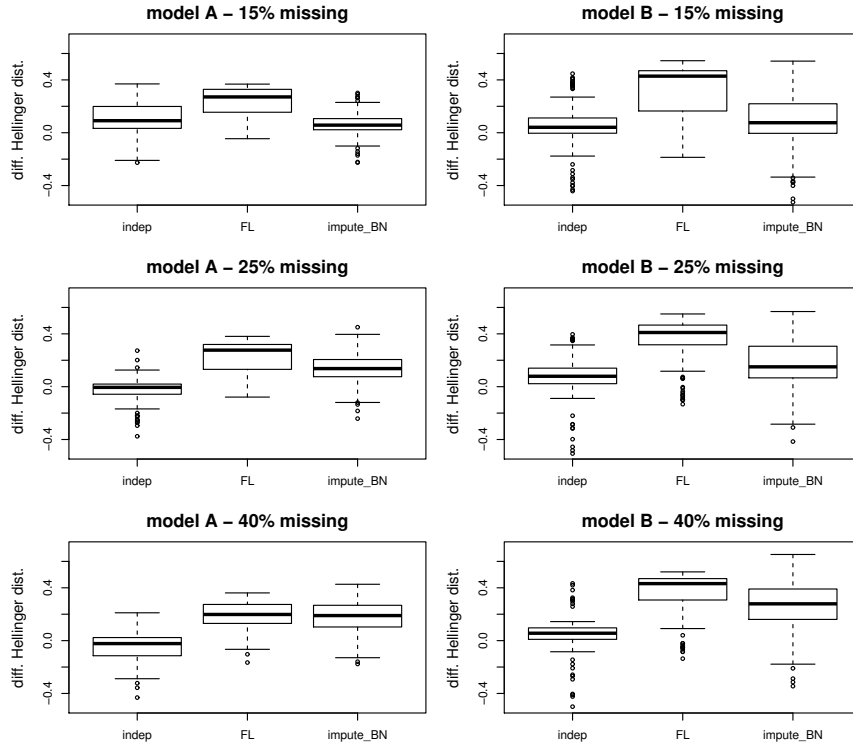
Figure 3: Boxplots of the difference between the Hellinger distance of each method with respect to our methodology based on Bayesian network. A positive difference means that our methodology performed better than the other method.

better or similar to the other methods (see Supplementary Table 3). The methods that impute with the mode (`indep_M`, `FL_M`, `impute_BN` and `BN_M`) performed generally poorer than the others and especially than `BN_E` (the MSE of `BN_E` was always significantly lower, see Supplementary Table 3). As expected from the results on the estimation of the joint distribution, in model A, when the dependencies between the variables are weaker, `indep_E` achieved always the lowest MSE. The method `BN_E` obtained a similar MSE in the case with 15% of missing values, while a greater error in the other cases but remaining among the best methods (see Supplementary Table 3). Furthermore, the method `FL_E` was always centered close to 0.25 with a variance close to 0, meaning that the missing values were mostly estimated with 0.5, which is not helpful for further analysis. In model B, `BN_E` generally gave a significantly more accurate imputation (only with 40% of missing it achieved an error similar to `indep_E` and with 25% similar to `FL_E`), since it better learned and employed all the information regarding the dependencies between the variables. For the sake of completeness, we also performed imputation by using a random draw from the conditional distribution estimated by `FL` and `BN` (data not shown). We obtained a very discouraging MSE, which was higher than the one achieved by the corresponding imputation with the mode and thus it would potentially lead to high errors in the survival tree estimation. Overall, `BN_E` and `indep_E` achieved the best performance, thus we use only these methods to impute the data in the subsequent analysis. In particular, `BN_E` outperformed all other methods in the presence of strong dependencies among the covariates (which cannot be known in advance in practice). We point out that we employed methods that impute missing data with the expected values because in our applications we have only binary, ordinal and/or discrete variables. Obviously, in presence of nominal variables with more than two categories, one should resort to imputation by the mode instead.

Afterwards, we tested if data imputation had improved the estimation of the classification scheme of the survival tree (which otherwise handles the problem of the missing data with surrogate splits, `SS`). In this
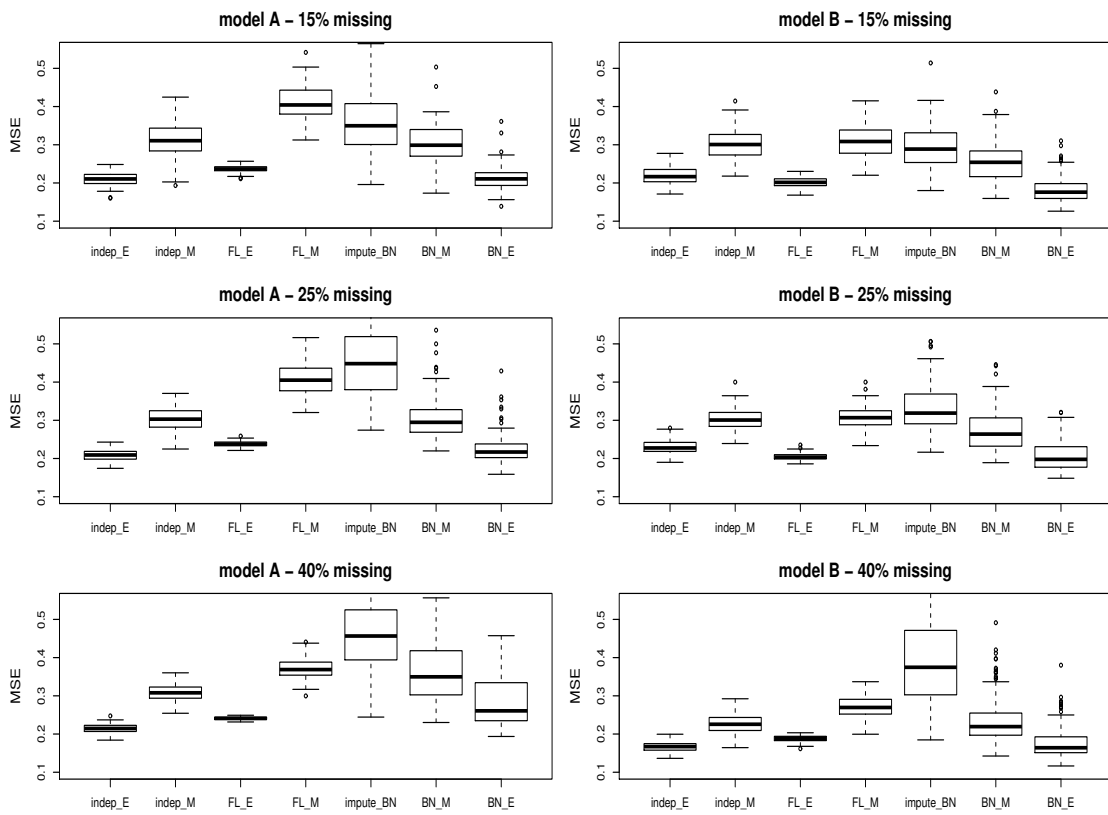
Figure 4: Boxplots of the mean square error (MSE) made in the imputation by several methods, in all scenarios.

case, we considered 18 possible scenarios for the training datasets, given by all combinations of dependency model, percentage of missing data and percentage of censoring data. We applied the survival tree to both the original training dataset (with the missing values) and the one obtained after imputing the data either with BN_E or with indep_E. The tree models so derived were then used to stratify the samples in the complete and uncensored test dataset corresponding to the same dependency model. For the evaluation, we employed the widely used Brier score (BS), which is a square error for survival prediction, accounting also for censoring observations [13]. Thus, the method with the lowest score denotes the one with the best performance. For a fixed time t, it is defined as

$$BS(t) = \frac{1}{n} \sum_{u=1}^{n} \left((0 - \widehat{S}(t|G_u))^2 \frac{\mathcal{I}_{\{Y_u \leq t, \delta_u = 1\}}}{\widehat{G}(Y_u)} + (1 - \widehat{S}(t|G_u))^2 \frac{\mathcal{I}_{\{Y_u > t\}}}{\widehat{G}(t)} \right),$$

where $\widehat{G}$ is the Kaplan-Meier estimate of the censoring distribution (i.e. considering as event $1 - \delta$) and the other variables are defined as in Section 3. In order to account for the whole survival estimation, it is possible to define even the integrated Brier score (IBS) by integrating the Brier score over the time with respect to some weight function. Its specification is

$$IBS = \frac{1}{t_{max}} \int_0^{t_{max}} BS(t) dt.$$

As suggested in Graf et al. [13], we compared the Brier scores at the median follow-up (i.e. median censoring time) of the training dataset, in order to have enough uncensored patients at risk at that moment for a reliable estimation of the survival prediction. For the same reason, the IBS was computed with t_{max} equal to the median follow-up. In the specific case of all uncensored data in the training dataset, we evaluated the Brier score at the median survival time and the IBS with $t_{max} = \max Y_u$. The function sbrier in the R package ipred [16] was employed to evaluate both scores. Figures 5 and 6 show, for all scenarios, the relative improvement of each score given by employing either BN_E or indep_E with respect to SS. The relative improvement for BN_E (and similarly for indep_E) is defined as in [16], that is, $(score_{SS} - score_{BN\_E})/score_{SS}$, where $score_{method}$ denotes the score (BS or IBS) obtained with $method$. Overall, the survival tree combined with the imputation of the data obtained better or similar results than without the imputation. In particular, when the covariates had weak dependencies, the imputation improved the estimation of the classification scheme (and thus the survival prediction) especially in presence of a high percentage of missing data. Instead, when the dependencies were strong, the rate of improvement in the survival prediction was sometimes smaller and it depended on the combination of the percentage of both missing and censored data, since the problem is harder to solve in this case. Moreover, in case of a strong dependency between the covariates, surrogate splits are better selected, making predictions of SS more precise. Comparing indep_E and BN_E, we observe that often the latter achieved a slightly higher relative improvement, especially in presence of 40% of missing data (which is a situation similar to the real practice).

In order to test whether the scores obtained by employing the imputation methods were usually better than the ones resulted from SS, we performed a paired Wilcoxon signed-rank one-sided test in all scenarios. In the same way, we also tested whether the performance of BN_E was better than the one of indep_E. As shown in Supplementary Table 4, apart from four cases, we were always able to reject the null hypothesis in favor that BN_E achieved a lower score than SS. The test was not significant for BS in the case of model A, with uncensored data and 15%–25% of missing data and with 25% of censored data and 15% of missing data (this latter case was also the only one not significant for IBS). In these scenarios, the p-value was not significant even in the corresponding two-sided test (thus, meaning that we cannot reject that the two methods behaved similarly). Regarding indep_E (see Supplementary Table 5), it behaved similar to SS also in other additional four cases in model A: for BS, in presence of 40% of missing data and 0%–25% of censored data, and for IBS, with uncensored data and 15%–25% of missing data. Even for indep_E, when the one-sided test was not significant also the corresponding two-sided test was not significant. In several scenarios (see Supplementary Table 6), BN_E achieved a significantly lower score than indep_E, and in all the other cases their scores were similar. BN_E outperformed indep_E especially with uncensored data and
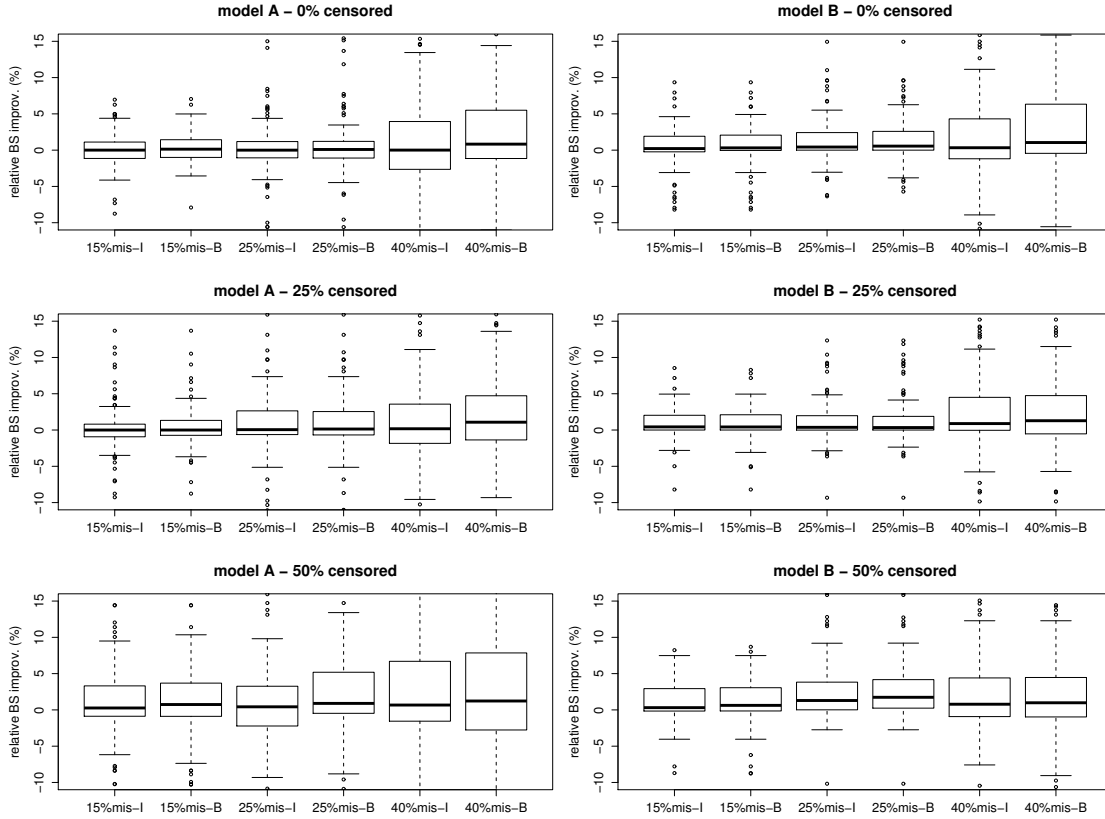
Figure 5: Boxplots of the relative improvement of the Brier score computed for the survival tree when applied after the imputation (obtained with either `indep_E` or `BN_E`) with respect to using the original data (that is, using `SS`), in all scenarios. A positive relative improvement indicates that the corresponding method performed better than `SS`. In the labels, I and B denote `indep_E` and `BN_E`, respectively.

40% of missing data (sometimes also 15%), and with 50% of censored data and 25% of missing data (for both models and both scores). In presence of uncensored data, the survival tree can obtain more precise predictions, allowing us to distinguish the eventually different accuracy in the imputation methods in the several scenarios of covariate missingness. When the percentage of missing data is low (such as 15%), `BN_E` can estimate the dependency model better than with high percentage, thus obtaining a missing data imputation which can perform better or similar to the one of `indep_E`. When the percentage of missing data is high (such as 40%), `indep_E` encounters great difficulties in estimating the missing values, while `BN_E` can rely on the more informative estimated dependency model, even if it is less accurate than in the case of a small percentage of missing data. Obviously, in both cases, the outperformance of `BN_E` depends also on the type of dependency structure among the covariates (which cannot be known in advance in practice). In case of an intermediate percentage of missing data (such as 25%), both methods show difficulties. Thus, it is much easier to discriminate the better performance of `BN_E` with respect to `indep_E` in situations that are hard for the survival tree analysis (such as the presence of a high percentage of censoring data). Finally, overall the imputation of missing data improved significantly the estimation of the survival tree with respect to the use of surrogate splits, and a better result was particularly obtained using `BN_E`.

4.2. Application to real data

In order to verify the performance of our proposed methodology (`BN_E`) in the analysis of real data, we consider the data of two important diseases: diffuse large B-cell lymphoma (DLBCL) and marginal zone
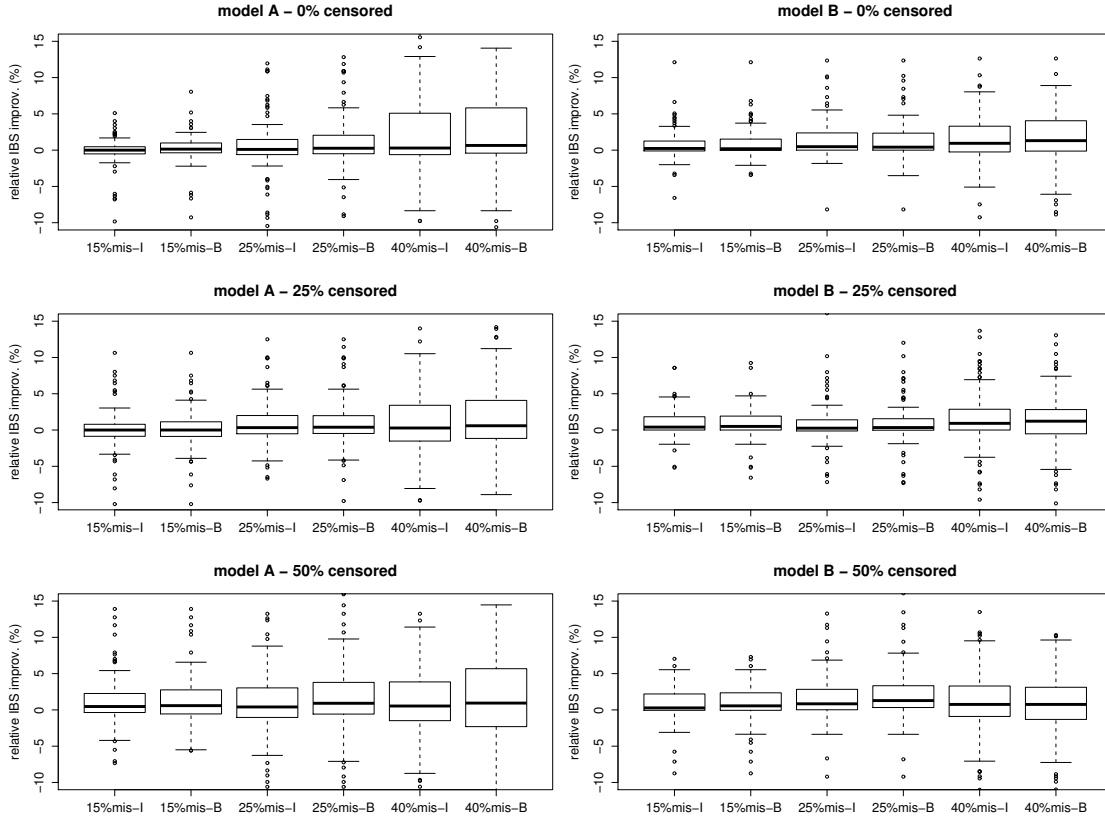
15

Figure 6: Boxplots of the relative improvement of the integrated Brier score computed for the survival tree when applied after the imputation (obtained with either `indep_E` or `BN_E`) with respect to using the original data (that is, using `SS`), in all scenarios. A positive relative improvement indicates that the corresponding method performed better than `SS`. In the labels, I and B denote `indep_E` and `BN_E`, respectively.

lymphoma (MZL). For each of them, we have two separate datasets: one for the training and one for the testing. In the case of DLBCL, the training dataset consists of 138 patients of Scandurra et al. [31] and the test one consists of 127 patients from the Oncology Institute of Southern Switzerland (IOSI). For DLBCL, we consider 18 binary and five discrete covariates assuming up to six distinct values each, and overall survival (OS) as survival outcome. The percentage of censored data is 67% and 71% in the training and in the test dataset, respectively. In the training dataset, the median and the maximum percentage of missing value per variable are 12% and 69%, respectively (0% and 56%, respectively, in the test dataset). Regarding the MZL data, the training dataset consists of 199 patients of Rinaldi et al. [29] and the test one consists of 180 patients of Zucca et al. [38]. In this case, we consider 15 binary and four discrete covariates assuming up to six distinct values each, and the OS outcome. Due to the type of disease, the percentage of censored data is 74% and 93% in the training and in the test dataset, respectively. In the training dataset, the median and the maximum percentage of missing value per variable are 24% and 59%, respectively (0% and 63%, respectively, in the test dataset).

For each disease, we imputed the missing values in the training data with `BN_E` and `indep_E`, and then we applied the survival tree to the two complete datasets, obtaining the corresponding classification scheme with a tree-like structure for prognostic patient stratification. We also applied the survival tree directly to the training dataset with missing data, in order to evaluate the eventual gain in accuracy in the survival prediction given by data imputation (with respect to using surrogate splits, `SS`). For this comparison, we applied all tree models to the corresponding test dataset for the classification of the patients and we

compared the Brier score at the median follow-up and the integrated Brier score until the median follow-up. The results in Table 3 show that imputing the missing data with BN_E in the training generally improves the estimation of the classification scheme given by the survival tree, since it always achieved a score smaller than SS. Unfortunately, the presence of many missing data and censored data in the test datasets affects the evaluation given by the score, and hence sometimes this difference is not significant at the paired Wilcoxon signed-rank test of the score computed per sample. Regarding indep_E, sometimes it achieved even a greater (that is, worse) score than SS and the score of BN_E per sample was always significantly smaller (that is, better) than indep_E. Moreover, by using the imputed data, the survival tree method was able to select important clinical variables while building the classification scheme, which were not chosen otherwise because of the high percentage of missing values. For example, in the case of DLBCL, one of the variables chosen in the tree, both using BN_E and indep_E, was hepatitis C serology. This variable is missing in 43% of the cases in the training dataset and in 56% in the test dataset. In the case of MZL, the presence of a high level of beta-2 microglobulin in the patient was one of the selected variables by BN_E, but it was missed by both indep_E and SS. This variable is missing in 59% of the cases in the training dataset and in 63% in the test dataset. Both variables, and especially beta-2 microglobulin, have been associated to a different outcome in the literature, supporting the results given by our methodology. The high percentage of missing values in these variables in the test patients turns the task of survival prediction into a very hard problem.

Table 3: BS at the median follow-up and IBS (in [0, median follow-up]) computed on the test datasets using the patient stratification obtained in the corresponding training datasets. The same scores were also computed for the single patients in order to perform a paired Wilcoxon signed-rank one-sided test, for testing that the corresponding score obtained by employing the first method indicated in the comparison is lower than the one resulted from the second method.

disease	score	method	value (median [IQR] per patient)	comparison	p-value
DLBCL	BS	SS	0.1676 (0.0364 [0;0.1921])	BN_E vs SS	0.0417
DLBCL	BS	BN_E	0.1467 (0.0042 [0;0.1615])	indep_E vs SS	0.9421*
DLBCL	BS	indep_E	0.1754 (0.0301 [0;0.2778])	BN_E vs indep_E	0.0149
DLBCL	IBS	SS	0.1167 (0.0256 [0.0044;0.1987])	BN_E vs SS	0.1029*
DLBCL	IBS	BN_E	0.1057 (0.03045 [0.0014;0.1668])	indep_E vs SS	0.9504*
DLBCL	IBS	indep_E	0.1197 (0.0453 [0.0045;0.1633]))	BN_E vs indep_E	0.0100
MZL	BS	SS	0.0727 (0 [0;0.0251])	BN_E vs SS	0.4872*
MZL	BS	BN_E	0.0699 (0 [0;0.0021])	indep_E vs SS	0.3369*
MZL	BS	indep_E	0.0664 (0 [0;0.0035])	BN_E vs indep_E	0.0158
MZL	IBS	SS	$0.0298\ (8.85\cdot10^{-4}\ [8.10\cdot10^{-5};0.0031])$	BN_E vs SS	<0.0001
MZL	IBS	BN_E	$0.0278\ (6.83\cdot10^{-5}\ [0;0.0002])$	indep_E vs SS	<0.0001
MZL	IBS	indep_E	$0.0281\ (2.56\cdot10^{-4}\ [1.84\cdot10^{-5};0.0003]))$	BN_E vs indep_E	<0.0001

*nonsignificant p-value at the corresponding two-sided test.

5. Conclusion

The treatment of missing data is a crucial issue in retrospective clinical studies, due to the often relatively small sample size. If not handled properly, it might affect results and reduce the accuracy of the analysis. Since the way of addressing this issue may depend on the particular application, in this paper we focus our attention on the case of prognostic patient stratification with survival trees. For this purpose, we suggest to impute missing values in order to apply the survival tree to a complete dataset. This is achieved by a framework that uses Bayesian networks, which are suitable for datasets where the number of missing data is considerably high and the amount of samples small, often the case in clinical studies. The Bayesian network is learned using a structural expectation-maximization procedure, where the maximization step is implemented with a globally optimal method, considering only the data of the covariates. This procedure is experimentally shown to outperform widely used techniques for data imputation on settings of simulated clinical data.

Using both simulated and real data of two distinct lymphomas, we also show that this methodology for data imputation may improve the patient stratification given by the survival tree, that is, patients can be divided in more precise groups (defined on the grounds of some clinical parameters), leading to more accurate predictions of the survival time. Moreover, in the real application, we also point out that the survival tree combined with our methodology for data imputation is able to select important clinical parameters even when they have a high percentage of missing values. These variables would not be selected without the imputation simply because of their missingness, even if it is known that they may influence the survival outcome. Therefore, the use of data imputation using Bayesian networks in real clinical studies might lead to an improvement in the survival prediction of patients and in the identification of prognostic factors.

Acknowledgements

References

[1] Breiman, L., Friedman, J. H., Olshen, R. A., Stone, C. J., 1984. Classification and regression trees. Wadsworth & Brooks, Monterey, California.

[2] Buntine, W., 1991. Theory refinement on Bayesian networks. In: D'Ambrosio, B. D., Smets, P., Bonissone, P. P. (Eds.), UAI-92. Morgan Kaufmann, San Francisco, CA, pp. 52–60.

[3] Ciampi, A., Thiffault, J., 1986. Stratification by stepwise regression, correspondence analysis and recursive partition: a comparison of three methods of analysis for survival data with covariates. Computational Statistics & Data Analysis 4 (3), 185204.

[4] Cooper, G. F., Herskovits, E., 1992. A Bayesian method for the induction of probabilistic networks from data. Machine Learning 9, 309–347.

[5] Cox, D., 1972. Regression models and life-tables. Journal of the Royal Statistical Society. Series B 34 (2), 187–220.

[6] Cussens, J., 2011. Bayesian network learning with cutting planes. In: Proceedings of the Twenty-Seventh Annual Conference on Uncertainty in Artificial Intelligence (UAI-11). pp. 153–160.

[7] Davis, R. B., Anderson, J. R., 1989. Exponential survival trees. Statistics in Medicine 8, 947–961.

[8] de Campos, C. P., Ji, Q., 2010. Properties of Bayesian Dirichlet scores to learn Bayesian network structures. In: AAAI Conference on Artificial Intelligence. AAAI Press, pp. 431–436.

[9] de Campos, C. P., Ji, Q., Mar 2011. Efficient structure learning of Bayesian networks using constraints. Journal of Machine Learning Research 12, 663–689.

[10] Di Zio, M., Scanu, M., Coppola, L., Luzi, O., Ponti, A., 2004. Bayesian networks for imputation. Journal of the Royal Statistical Society A 167(2), 309–322.

[11] Fana, F., Nunnb, M., Su, X., 2009. Multivariate exponential survival trees and their application to tooth prognosis. Computational Statistics & Data Analysis 53 (4), 11101121.

[12] Friedman, N., 1998. The Bayesian structural EM algorithm. In: Cooper, G. F., Moral, S. (Eds.), Conference on Uncertainty in Artificial Intelligence. Morgan Kaufmann, San Francisco, CA, pp. 129–138.

[13] Graf, E., Schmoor, C., Sauerbrei, W., Schumacher, M., 1999. Assessment and comparison of prognostic classification schemes for survival data. Statistics in Medicine 18, 2529–2545.

[14] Heckerman, D., Geiger, D., Chickering, D. M., 1995. Learning Bayesian networks: the combination of knowledge and statistical data. Machine Learning 20 (3), 197–243.

[15] Hothorn, T., Hornika, K., Zeileis, A., 2006. Unbiased recursive partitioning: A conditional inference framework. Journal of Computational and Graphical Statistics 15 (3), 651–674.

[16] Hothorn, T., Lausen, B., Benner, A., Radespiel-Tröger, M., 2004. Bagging survival trees. Statistics in Medicine 23, 77–91.

[17] Jaakkola, T., Sontag, D., Globerson, A., Meila, M., 2010. Learning Bayesian network structure using LP relaxations. In: Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics. pp. 358–365.

[18] Keleş, S., Segal, M. R., 2002. Residual-based tree-structured survival analysis. Statistics in Medicine 21, 313–326.

[19] Koller, D., Friedman, N., 2009. Probabilistic Graphical Models. MIT press.

[20] LeBlanc, M., Crowley, J., 1992. Relative risk trees for censored survival data. Biometrics 48, 411–425.

[21] LeBlanc, M., Crowley, J., 1993. Survival trees by goodness of split. Journal of the American Statistical Association 88 (422), 457–467.

[22] Little, R. J. A., Rubin, D. B., 1987. Statistical analysis with missing data. John Wiley & Sons, New York.

[23] Meila, M., Jordan, M., 1998. Estimating dependency structure as a hidden variable. In: Conference on Advances in Neural Information Processing Systems. pp. 584–590.

[24] Nilsson, D., 1998. An efficient algorithm for finding the M most probable configurations in Bayesian networks. Statistics and Computing 9, 159–173.

[25] Peto, R., Peto, J., 1972. Asymptotically efficient rank invariant test procedures. Journal of the Royal Statistical Society. Series A 135 (2), 185–207.

[26] Ramoni, M., Sebastiani, P., 1997. Learning Bayesian networks from incomplete databases. In: Proceedings of the Thirteenth Conference on Uncertainty in Artificial Intelligence. pp. 401–408.

[27] Riggelsen, C., 2006. Learning Bayesian networks from incomplete data: An efficient method for generating approximate predictive distributions. In: Proceedings of the 2006 SIAM International Conference on Data Mining. SIAM, p. 130.

[28] Riggelsen, C., Feelders, A., 2005. Learning Bayesian network models from incomplete data using importance sampling. In: Cowell, R. G., Ghahramani, Z. (Eds.), Proc. of AI & Statistics. Society for Artificial Intelligence and Statistics, pp. 301–308, (Available electronically at http://www.gatsby.ucl.ac.uk/aistats/).

[29] Rinaldi, A., Mian, M., et al., E. C., 2011. Genome wide DNA-profiling of marginal zone lymphomas identifies subtype-specific lesions with an impact on the clinical outcome. Blood 117 (5), 1595–1604.

[30] Romero, V., Salmerón, A., 2004. Multivariate imputation of qualitative missing data using Bayesian networks. In: Soft methodology and random information systems. Springer, pp. 605–612.

[31] Scandurra, M., Mian, M., et al., T. G., 2010. Genomic lesions associated with a different clinical outcome in diffuse large B-cell lymphoma treated with R-CHOP-21. British Journal of Heamatology 151 (3), 221–231.

[32] Schauerhuber, M., Zeileis, A., Meyer, D., Hornik, K., 2008. Benchmarking open-source tree learners in R/RWeka. In: Preisach, C., Burkhardt, H., Schmidt-Thieme, L., Decker, R. (Eds.), Data Analysis, Machine Learning and Applications (Proceedings of the 31st Annual Conference of the Gesellschaft fr Klassifikation e.V., Albert-Ludwigs-Universitt Freiburg, March 7-9, 2007). Springer-Verlag, pp. 389–396.

[33] Segal, M. R., 1988. Regression trees for censored data. Biometrics 44 (1), 35–47.

[34] Silander, T., Myllymaki, P., 2006. A simple approach for finding the globally optimal Bayesian network structure. In: 22nd Conference on Uncertainty in Artificial Intelligence. AUAI Press, Arlington, Virginia, pp. 445–452.

[35] Simon, N., Friedman, J., Hastie, T., Tibshirani, R., 2011. Regularization paths for Cox's proportional hazards model via coordinate descent. Journal of Statistical Software 39 (5).

[36] Singh, M., 1998. Learning Bayesian networks for solving real-world problems. Ph.D. thesis, University of Pennsylvania.

[37] Tanner, M. A., Wong, W. H., Jun 1987. The calculation of posterior distributions by data augmentation. Journal of the American Statistical Association 82, 528–540.

[38] Zucca, E., Conconi, A., Pedrinis, E., Cortelazzo, S., Motta, T., Gospodarowicz, M., Patterson, B., Ferreri, A., Ponzoni, M., Devizzi, L., Giardini, R., Pinotti, G., Capella, C., Zinzani, P., Pileri, S., López-Guillermo, A., Campo, E., Ambrosetti, A., Baldini, L., Cavalli, F., 2003. Nongastric marginal zone B-cell lymphoma of mucosa-associated lymphoid tissue. Blood 101 (7), 2489–2495.

Supplementary Material for Bayesian network data imputation with application to survival tree analysis by Paola M.V. Rancoita, Marco Zaffalon, Emanuele Zucca, Francesco Bertoni and Cassio P. de Campos

Supplementary tables

Table 1: P-value of the paired Wilcoxon signed-rank one-sided test, for testing the alternative hypothesis that the Hellinger distance obtained by employing our methodology is lower than the one resulted from the other methods. The test was performed in all scenarios.

	model A			model B		
method	15% miss	25% miss	40% miss	15% miss	25% miss	40% miss
indep	<0.0001	0.9822**	1.0000**	0.0003	<0.0001	<0.0001
FL	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001
impute_BN	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001

**significant p-value at the corresponding two-sided test.

Table 2: P-value of the paired Wilcoxon signed-rank one-sided test, for testing the alternative hypothesis that the Kullback-Leibler divergence obtained by employing our methodology is lower than the one resulted from the other methods. The test was performed in all scenarios.

	model A			model B		
method	15% miss	25% miss	40% miss	15% miss	25% miss	40% miss
indep	<0.0001	0.9995**	1.0000**	0.0029	<0.0001	0.0029
FL	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001
impute_BN	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001

**significant p-value at the corresponding two-sided test.

Table 3: P-value of the paired Wilcoxon signed-rank one-sided test, for testing the alternative hypothesis that the MSE of the imputation obtained by employing BN_E is lower than the one resulted from the other methods. The test was performed in all scenarios.

	model A			model B		
method	15% miss	25% miss	40% miss	15% miss	25% miss	40% miss
indep_E	0.4774*	1.0000**	1.0000**	<0.0001	<0.0001	0.6670*
indep_M	<0.0001	<0.0001	0.0001	<0.0001	<0.0001	<0.0001
FL_E	<0.0001	<0.0001	1.0000**	<0.0001	0.3444*	0.0008
FL_M	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001
impute_BN	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001
BN_M	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001

*nonsignificant and **significant p-value at the corresponding two-sided test.

Table 4: P-value of the paired Wilcoxon signed-rank one-sided test, for testing the alternative hypothesis that the score obtained by employing BN_E is lower than the one resulted from SS. The test was performed, in all scenarios, for both the Brier score (BS) and the integrated Brier score (IBS) as described in the text.

		model A			model B		
score	% censored	15% miss	25% miss	40% miss	15% miss	25% miss	40% miss
BS	0	0.0951*	0.1532*	0.0050	0.0010	<0.0001	0.0004
BS	25	0.1829*	0.0189	0.0024	<0.0001	0.0001	0.0007
BS	50	0.0081	0.0019	0.0214	0.0003	<0.0001	0.0128
IBS	0	0.0338	0.0050	0.0001	<0.0001	<0.0001	<0.0001
IBS	25	0.5272*	0.0040	0.0024	<0.0001	0.0004	<0.0001
IBS	50	0.0011	0.0007	0.0167	0.0001	<0.0001	0.0242

*nonsignificant p-value at the corresponding two-sided test.

Table 5: P-value of the paired Wilcoxon signed-rank one-sided test, for testing the alternative hypothesis that the score obtained by employing indep_E is lower than the one resulted from SS. The test was performed, in all scenarios, for both the Brier score (BS) and the integrated Brier score (IBS) as described in the text.

		model A			model B		
score	% censored	15% miss	25% miss	40% miss	15% miss	25% miss	40% miss
BS	0	0.4609*	0.3228*	0.1813*	0.0032	<0.0001	0.0050
BS	25	0.3579*	0.0318	0.0754*	<0.0001	0.0001	0.0007
BS	50	0.0176	0.0444	0.0173	0.0004	<0.0001	0.0104
IBS	0	0.4490*	0.0597*	0.0073	0.0001	<0.0001	<0.0001
IBS	25	0.6880*	0.0076	0.0196	<0.0001	0.0007	0.0006
IBS	50	0.0019	0.0256	0.0269	<0.0001	<0.0001	0.0106

*nonsignificant p-value at the corresponding two-sided test.

Table 6: P-value of the paired Wilcoxon signed-rank one-sided test, for testing the alternative hypothesis that the score obtained by employing BN_E is lower than the one resulted from indep_E. The test was performed, in all scenarios, for both the Brier score (BS) and the integrated Brier score (IBS) as described in the text.

		model A			model B		
score	% censored	15% miss	25% miss	40% miss	15% miss	25% miss	40% miss
BS	0	0.1495*	0.1697*	0.0240	0.0779*	0.2812*	0.0486
BS	25	0.2641*	0.4452*	0.0089	0.2655*	0.4511*	0.2191*
BS	50	0.1173*	0.0332	0.0734*	0.2558*	0.0008	0.9137*
IBS	0	0.0164	0.0968*	0.0157	0.0420	0.6947*	0.0358
IBS	25	0.2786*	0.5582*	0.0828*	0.4814*	0.4061*	0.3308*
IBS	50	0.1286*	0.0238	0.0860*	0.2194*	0.0007	0.9326*

*nonsignificant p-value at the corresponding two-sided test.
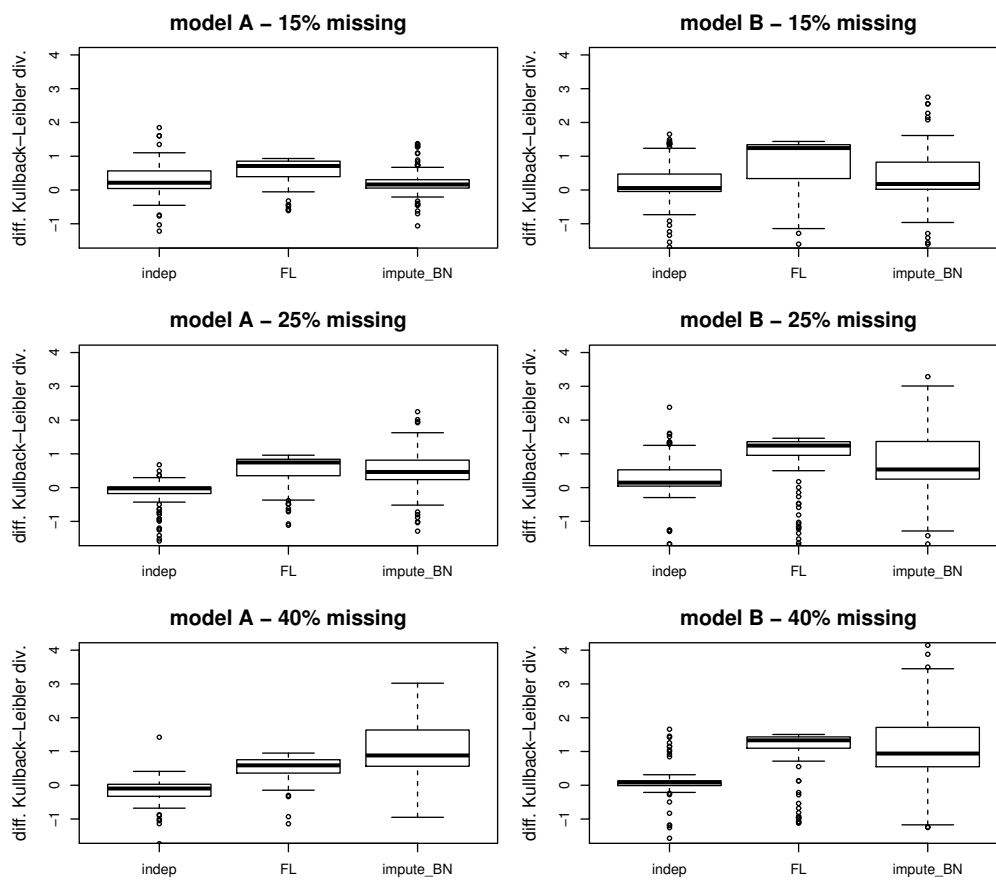
Supplementary figures

Figure 1: Boxplots of the difference between the Kullback-Leibler divergence of each method with respect to our methodology based on the Bayesian network. A positive difference means that our methodology performed better than the other method.