Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17)

Learning Bayesian Networks with Incomplete Data by Augmentation

Tameem Adel
University of Manchester, UK
tameem.hesham@gmail.com

Cassio P. de Campos
Queen's University Belfast, UK
c.decampos@qub.ac.uk

Abstract

We present new algorithms for learning Bayesian networks from data with missing values using a data augmentation approach. An exact Bayesian network learning algorithm is obtained by recasting the problem into a standard Bayesian network learning problem without missing data. As expected, the exact algorithm does not scale to large domains. We build on the exact method to create an approximate algorithm using a hill-climbing technique. This algorithm scales to large domains so long as a suitable standard structure learning method for complete data is available. We perform a wide range of experiments to demonstrate the benefits of learning Bayesian networks with such new approach.

Introduction

Missing entries in real-world data exist due to various reasons. For instance, it can be due to damage of the device used to record feature values; a metal detector might fail to produce a signal denoting the existence of a metal due to a certain malfunction. Results can be incomplete in an industrial experiment due to mechanical breakdowns not necessarily related to the performed experiment (Little and Rubin 1987). Recommendation data can have missing values since participants in the recommendation system did not rate all the available songs, films, books, etc. While data missingness in the above examples can mostly be assumed to be generated by a random process which depends only on the observed data, usually referred to as *missing at random (MAR)* (Little and Rubin 1987), this assumption might fail in other examples. People seeking for health insurance might refuse to give an answer to certain questions in order to reduce the costs, e.g. 'do you smoke?', and in many cases this can be seen as an indication of one specific answer. In such cases we say that data are *missing not at random*, or MNAR (see for instance (Van den Broeck et al. 2014)).

Given a dataset with categorical random variables, the Bayesian network structure learning problem refers to finding the best network structure (a directed acyclic graph, or DAG) according to a score function based on the data (Heckerman, Geiger, and Chickering 1995). As well known, learning a Bayesian network from complete data is NP-complete (Chickering 1996), and the task becomes even harder with incomplete data. In spite of that, the problem of learning a Bayesian network from incomplete data by (an optimistic) augmentation belongs to the same complexity class, as we will show later on. Because of such result, we investigate and obtain a new exact algorithm for the problem, based on reformulating it into a standard structure learning without missing data. This is the first exact algorithm for the problem, to the best of our knowledge. In contrast to previous work, our algorithm performs both tasks, namely structure learning and data imputation, in a single shot rather than learning the Bayesian network and then dealing with the missing data, possibly in an iterative manner (Friedman 1998). Based on the optimization that is required to solve the problem and on the exact algorithm, we devise a hill-climbing approximate algorithm. The hill-climbing regards the completions of the missing values only, while the structure optimization is performed by any off-the-shelf algorithm for structure learning under complete data.

Most previous work to learn the structure of Bayesian networks from incomplete data has focused on MAR. The seminal algorithm of Friedman (1998) introduced an iterative method based on the Expectation-Maximization (EM) technique, referred to as structural EM. Implementation of structural EM begins with an initial graph structure, followed by steps where the probability distribution of variables with missing values is estimated by EM, alternated with steps in which the expectation of the score of each neighbouring graph is computed. After convergence, the graph maximizing the score is chosen. Many other algorithms have used ideas from structural EM and deal separately with the missing values and the structure optimization using complete data (Borchani, Amor, and Mellouli 2006; Leray and Francois 2005; Meila and Jordan 1998; Ramoni and Sebastiani 1997; Riggelsen 2006; Riggelsen and Feelders 2005). In (Rancoita et al. 2016), structures are learned from incomplete data using a structural EM whose maximization step is performed by an anytime method, and the 'expectation' step imputes the missing values using expected means, or modes, of the current estimated joint distribution. By using modes in each iteration (Ramoni and Sebastiani 1997), the EM method is sometimes called *hard* EM, and is close to our work. In some sense, we work with a global optimization version of hard EM. While this is not exactly considering data to be MNAR,

such approach fits less the observed data and performs well for MNAR missing data when compared to structural EM, as we will empirically show. We emphasize that the actual missingness process is not disclosed to the methods and is not assumed to be somehow known, and that we are mainly interested in structure learning. Given the difficulties of structure learning itself, we assume that the underlying distribution is identifiable (in short terms, provided enough data are available, one could reconstruct such distribution, see for instance (Mohan, Pearl, and Tian 2013)).

We perform experiments on a set of heterogeneous datasets. We base the evaluation on imputation accuracy in its pure form, as well as in the forms of classification accuracy and semi-supervised learning accuracy. Experiments show the improvements achieved by the proposed algorithms in all scenarios. Regarding the comparison between our exact and approximate methods, experiments suggest that accuracy levels achieved by the approximate algorithm are close to those achieved by the optimal learning algorithm, with the former being much faster and scalable.

Bayesian Network Structure Learning

Let $\mathbf{X} = (X_1, \ldots, X_m)$ refer to a vector of categorical random variables, taking values in $\mathrm{O}_{\mathbf{X}} = \times_i \mathrm{O}_{X_i}$, where $\mathrm{O}_{\mathbf{X}}$ represents the Cartesian product of the state space, O_{X_i}, of each X_i. Denote by $\mathcal{D}$ an n-instance dataset where each instance $\mathbf{D}_u = (d_{u,1}, d_{u,2}, \ldots, d_{u,m})$ is such that $d_{u,i}$ is either an observed value $o_{u,i} \in O_{X_i}$ or a special symbol denoting the entry is missing. Let $\mathbf{Z}_u$ denote a completion for variables with missing values in instance u and $z_{u,i}$ for the missing value of X_i.

A Bayesian network, $\mathcal{M}$, is a probabilistic graphical model based on a structured dependency among random variables to represent a joint probability distribution in a compact and tractable manner. Here, it represents a joint probability distribution $\Pr_{\mathcal{M}}$ over a collection of categorical random variables, $\mathbf{X}$. We define a Bayesian network 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 $\Pr_{\mathcal{M}}(X_i|PA_i)$ 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 $\Pr_{\mathcal{M}}(X_1, \ldots, X_m) = \prod_i \Pr_{\mathcal{M}}(X_i|PA_i)$. We define $r_i \geq 2$ as the number of values in O_{X_i}, i.e. $r_i = |O_{X_i}|$, and r_{PA_i} as the number of possible realizations of the parent set, that is, $r_{PA_i} = \prod_{X_l \in PA_i} r_l$. Let $R = \max_i r_i$.

Given a complete dataset $\mathcal{D}$ with n instances, 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 $\mathbf{X}$. We consider here the score function $s_{\mathcal{D}}$ to be the Bayesian Dirichlet Equivalent Uniform (BDeu) criterion (Buntine 1991; Cooper and Herskovits 1992) (other decomposable scores could be used too), so we have $s_{\mathcal{D}}(\mathcal{G}) = \sum_i s_{\mathcal{D}}(X_i, PA_i)$. We however have to deal with the missing part of the data, which we treat by completing the missing values in the best possible way (an optimistic completion):

$$(\mathcal{G}^*, \mathcal{Z}^*) = \operatorname*{argmax}_{\mathcal{G} \in \boldsymbol{\mathcal{G}},\ \mathcal{Z} \in \boldsymbol{\mathcal{Z}}} s_{\mathcal{D}}(\mathcal{G}, \mathcal{Z}) = \operatorname*{argmax}_{\mathcal{G} \in \boldsymbol{\mathcal{G}},\ \mathcal{Z} \in \boldsymbol{\mathcal{Z}}} \sum_i s_{\mathcal{D}}(X_i, PA_i; \mathbf{Z}_{\{X_i\} \cup PA_i}) \quad (1)$$

where $\boldsymbol{\mathcal{Z}} = \times_u O_{\mathbf{Z}_u}$ and $s_{\mathcal{D}}(\mathcal{G}, \mathcal{Z})$ is the score $s_{\mathcal{D}}(\mathcal{G})$ evaluated for the complete data when its missing values are replaced by $\mathcal{Z}$, while $s_{\mathcal{D}}(X_i, PA_i; \mathbf{Z}_{\{X_i\} \cup PA_i})$ is the local score for a node X_i with parent set PA_i (note that such computation only depends on the completion $\mathbf{Z}_{\{X_i\} \cup PA_i}$ of the involved variables). We refer to this optimization task as the structure learning problem by optimistic augmentation. It can be applied to MAR data, but we argue that it is particularly suitable to MNAR when compared to the standard techniques such as structural EM. From the optimization viewpoint, this can be seen as a global optimization approach to *hard* EM, since we complete the data with their mode, but we do it globally instead of in an iterative process such as EM. As well known, *hard* EM can be seen as a subcase of EM, since it is equivalent to allowing EM to use only *degenerate* mass functions in its expectation step.

Theorem 1. *The decision version associated to the structure learning problem by optimistic augmentation is NP-complete.*

Proof. Hardness is obtained by realizing that this problem generalizes the structure learning problem without missing data, which is NP-hard (Chickering 1996). Pertinence in NP holds since given $\mathcal{G}$ and $\mathcal{Z}$, the score function $s_{\mathcal{D}}$ can be computed in polynomial time. □

Since the problem is a combinatorial optimization over a discrete domain (both DAGs and completions of data are discrete entities), we could resort to enumerating all possible solutions. This is obviously infeasible for both: the number of DAGs grows super-exponentially in the number of variables and the number of completions grows exponentially in the number of missing values. We will now present an exact algorithm for the problem which transforms it into a standard structure learning problem, and later we modify the approach to perform approximate learning. In this respect, we define as a t-local optimal solution for Equation (1) a pair $(\mathcal{G}, \mathcal{Z})$ such that $s_D(\mathcal{G}, \mathcal{Z}) \geq s_D(\mathcal{G}', \mathcal{Z}')$ for all $\mathcal{G}'$ and all $\mathcal{Z}'$ with $\mathrm{HD}(\mathcal{Z}, \mathcal{Z}') \leq t$, where HD is the Hamming distance, that is, $(\mathcal{G}, \mathcal{Z})$ is optimal with respect to any other pair whose completion of the data has at most t elements different from $\mathcal{Z}$. A global optimal solution is a ∞-local optimal solution.

Optimal (Exact) Learning Algorithm

We assume that a standard structure learning algorithm for complete data is available to us, which is based on the framework of two main optimizations: (i) parent set identification and (ii) structure optimization. Step (i) concerns building a list of candidate parent sets for each variable, while Step (ii) optimizes the selection of a parent set for each variable in a way to maximize the total score while ensuring that the graph is a DAG. This latter step can be tackled by

exact or approximate methods (Bartlett and Cussens 2013; Scanagatta et al. 2015) (in our experiments we will employ an exact method such that we are sure that the quality of results is only affected/related to the proper treatment of the missing data, but for very large domains any approximate method could be used too).

The exact algorithm for solving Equation (1) is based on modifying the parent set identification step. This step has no known polynomial-time solution if we do not impose a maximum number of parents (Koivisto 2006), so we will assume that such a bound k is given. We compute the candidate list by using one of the available approaches (de Campos and Ji 2011; Scanagatta et al. 2015) to guide the search, but for each candidate to be evaluated, the corresponding variables in the dataset might contain missing values. The first part of the transformation is to create gadgets composed of some new artificial variables which will be related to the missing values and will enable the inclusion of all possible replacements of missing values by augmenting the original domain.

Over the whole dataset, for each and every missing value, let us denote it by (u, i) for sample u and variable X_i, we include artificial variables $X_{(u,i),1}, \ldots, X_{(u,i),r_i}$. Each $X_{(u,i),j}$ has two parent set candidates: (i) $\mathbf{X} \cup \{X_{(u,i),1+(j \bmod r_u)}\}$ with score zero (assuming all other score values are negative, without loss of generality) and (ii) $\emptyset$ with score $-\lambda$, with λ a large enough value (e.g. greater than the sum of all other absolute scores). We further illustrate the idea via an example for variable X_1 with $r_1 = 3$: Assume $m = 3$, $r_1 = 3$ and there is one missing value at $(u, 1)$. An artificial variable is included for each possible completion $z_{u,1}$, resulting in a total of three new variables, $X_{(u,1),1}, X_{(u,1),2}, X_{(u,1),3}$. The following gadget, consisting of two parent set candidates per artificial variable, is added to the list of parent set scores (we know that only one parent set per variable will be chosen during the optimization phase later on):

$$
\begin{aligned}
&s(X_{(u,1),1}, \{X_{(u,1),2}, X_1, X_2, X_3\}) = 0,\\
&s(X_{(u,1),1}, \emptyset) = -\lambda,\\
&s(X_{(u,1),2}, \{X_{(u,1),3}, X_1, X_2, X_3\}) = 0,\\
&s(X_{(u,1),2}, \emptyset) = -\lambda,\\
&s(X_{(u,1),3}, \{X_{(u,1),1}, X_1, X_2, X_3\}) = 0,\\
&s(X_{(u,1),3}, \emptyset) = -\lambda.
\end{aligned}
$$

According to this gadget, each artificial variable will either have no parent variables or all other original variables as well as one other artificial variable as its set of parents. The case with no parents leaves open the opportunity to choose the variable representing such completion as a potential parent for all original variables. In contrast, the cases with all variables as parents disables such completion from being chosen as a parent by the original variables, otherwise it would create a cycle. Due to including one artificial variable as a parent of the next artificial variable, at least one parent set among those with score zero cannot be chosen (otherwise a cycle is formed), and because they are all very good scores when compared to $-\lambda$, all but one will certainly be chosen. There is one such gadget per missing value in the original dataset, so we spend time $O(R \cdot m \cdot C)$, where C is the number of missing values.

Finally, we return to the computation of the score for a given variable and parent set. Let X_i be the variable of interest and $PA_i = \{X_{i_1}, \ldots, X_{i_q}\}$ for which the score must be evaluated. At this moment, we consider all possible completions $\mathbf{Z}_{\{X_i\} \cup PA_i}$ and compute the scores $s_{\mathcal{D}}(X_i, PA_i; \mathbf{Z}_{\{X_i\} \cup PA_i})$ for each one of them. In order to reduce the problem to a standard structure learning without missing data, we must index these scores somehow. This is made possible via the new artificial variables:

$$
\begin{aligned}
&s_{\mathcal{D}}(X_i, PA_i; \mathbf{Z}_{\{X_i\} \cup PA_i}) =\\
&\quad s_{\mathcal{D}}(X_i, PA_i \cup \{X_{(u,i),z_{u,j}} : z_{u,j} \in \mathbf{Z}_{\{X_i\} \cup PA_i}\}),
\end{aligned}
$$

that is, for each imputed missing value $z_{u,j}$ appearing for variable X_i or PA_i we will have an extra parent within the parent set that tells which completion was used for that missing value, according to the completion $\mathbf{Z}_{\{X_i\} \cup PA_i}$. This idea is applied to every evaluation of the score of a parent set, for every possible completion $\mathbf{Z}_{\{X_i\} \cup PA_i}$, so the final list of candidates will include only parent sets for which the completion of the data is 'known' at the time that the score is computed. In order to ensure that the completions are compatible among different local score computations, the gadgets explained before are enough, since they force that a certain completion be chosen for each missing value.

Theorem 2. *The exact algorithm transforms the structure learning problem by augmentation into a standard structure learning without missing data in time $O(R \cdot m \cdot C)$, plus time $O(n \cdot k \cdot R^c)$ per parent set evaluation, where C is the total number of missing values and c is the maximum number of missing values appearing in the variable of interest or in variables in the parent set being evaluated (hence polynomial in all parameters but c).*

There will be many score computations and entries in the list, exponential in the number of missing values involved. So the benefit of this approach is that usually only a few variables are involved in the score computation at the same time. The drawback is that it cannot handle datasets with many missing values for the same variable, since it is R^c times slower than the corresponding parent set evaluation without missing data. Next we address this issue by proposing an approximate method (the exact method is nevertheless useful in small domains and also important to check whether the approximate version achieves reasonable results).

Approximate Algorithm

Albeit locally to the variables involved in the evaluation of a parent set, the exact method considers all possible completions of the data. This is fine with a few missing values per variable, but if there are many missing values, in particular within the same variable, the exact method becomes computationally infeasible. We propose an approximate algorithm based on a hill-climbing idea. We start with an initial guess $\mathcal{Z}_0$ (or several different random guesses) for the completion of all missing values in the dataset. Then we execute the very same steps of the exact algorithm, but we restrict the

completions only to those which are at most t elements different from the current guess $\mathcal{Z}_h$. There are at most $(R \cdot m)^t$ completions $\mathcal{Z}'_h$ such that $\text{HD}(\mathcal{Z}_h, \mathcal{Z}'_h) \leq t$. We proceed as with the exact method, but applying such constraint during the transformation that was explained in the previous section. After the transformation is done, the structure optimization is run and a new structure and new data completion $\mathcal{Z}_{h+1}$ is obtained. We repeat the process until convergence, that is, until $\mathcal{Z}_{h+1} = \mathcal{Z}_h$.

Theorem 3. *The approximate algorithm transforms the structure learning problem by augmentation into a standard structure learning without missing data in time $O(R \cdot m \cdot C)$, plus time $O(n \cdot k \cdot (R \cdot m)^t)$ per parent set evaluation (C is the total number of missing values and t is the amount of locality of the approximation, as previously defined), that is, polynomial in all parameters but t.*

The outcome of the approximate learning algorithm is the network structure as well as the completion of all the missing data values. The approximate algorithm might lead to a locally optimal solution, but on the other hand it is much more scalable than the exact algorithm.

Theorem 4. *Provided that an optimal structure learning optimization algorithm is available, the approximate algorithm always converges to a t-local optimal solution.*

If we want to scale to very large domains, we could also resort to an approximate structure learning optimization algorithm (e.g. (Scanagatta et al. 2015)). In this case, our approximate algorithm could be used in domains with hundreds or even thousands of variables (using very small t), but we would lose the guarantee to converge to a t-local optimal solution (it would still be a local optimum, but we would have to define it locally also in terms of the graph structures).

Experiments

We perform experiments on simulated as well as real-world data. The main evaluation metric used is accuracy of the imputation of missing data values, either in the form of missing values spread throughout the data, or in the form of a binary classification problem where only the class variable can contain missing values. Most of our experiments are with binary data for the sake of exposition, even though the algorithms are general and can be used with any categorical data (as shown in the last experimental setting). To test significance, we perform a paired t-test with significance level at 5%. Throughout all tables of results, a result in **bold** refers to an accuracy value that is significantly better than its competitors, whereas showing two results belonging to the same experiment in **bold** means that each of them being significantly better than the rest of the competitors. For structure optimization, we use the exact solver referred to as Gobnilp (Bartlett and Cussens 2013) with the code available from https://www.cs.york.ac.uk/aig/sw/gobnilp/. We perform comparisons among the two proposed algorithms (exact and approximate) and the structural Expectation-Maximization (EM) algorithm (Friedman 1998). We compare accuracy of the three algorithms based on the percentage of correct imputations over all missing values. As for the structural EM, we have used the implementation available at https://github.com/cassiopc/csda-dataimputation (Rancoita et al. 2016). After convergence, we run the prediction of missing values using a *most probable explanation* query. We must emphasize that the task of Bayesian network structure learning with missing values is very challenging, since it is already challenging without missing values. Therefore, we have focused on real but controlled experiments where we can effectively run the algorithms and assert their quality. We use maximum number of parents, $k = 3$, and use $t = 1$.

Well-known Bayesian Networks

We perform experiments using real but small data sets in order to compare both exact and approximate algorithms. First, we employ the original Bayesian network model for Breast Cancer (Almeida et al. 2014), which contains 8 binary variables, we simulate 100 data instances. That model has been learned from cancer patients of the University of Wisconsin Medical Hospital. Features (Bayesian network nodes) include breast density, mass density, architectural distortion and others, in addition to the diagnosis variable whose binary value refers to benign or malignant (D'Orsi et al. 2003). We include two missing values per variable, resulting in a total of 16 missing values. These missing values are generated in a MNAR manner by randomly removing values that are equal to each other, that is, during the generation we enforce that all missing values are zero, or that all missing values are one. Imputation results of the proposed exact learning algorithm, approximate algorithm and structural EM are displayed in the first row of Table 1 over 100 repetitions of the experiment.

Second, we use the Bayesian network that has been learned from the Prostate Cancer data by the Tree Augmented Naive Bayes (TAN) (Friedman, Geiger, and Goldszmidt 1997), implemented by WEKA (Hall et al. 2009). The Prostate Cancer data were acquired during three different moments in time (Sarabando 2011; Almeida et al. 2014), i.e. during a medical appointment, after performing auxiliary exams, and five years after a radical prostatectomy. It contains 11 binary variables, and 100 instances are generated. We randomly produce two MNAR missing values per variable, resulting in a total of 22 missing values. Results are shown in the second row of Table 1.

Third, the well-known ASIA network is used (Lauritzen and Speigelhalter 1988). We generate 100 instances according to this model, which contains 8 binary variables. Two missing values per variable are randomly generated according to MNAR. Imputation results are displayed in the third row of Table 1. Results indicate that the algorithms proposed here are significantly better than structural EM. More interestingly, results of the proposed exact and approximate BN learning algorithms are not significantly different, which supports the use of the (more efficient) approximate method for larger domains.

(LUng CAncer Simple set) LUCAS Dataset

The LUCAS dataset contains data of the LUCAS causal Bayesian network (Fogelman-Soulie 2008) with 11 binary variables, as well as the binary class variable, and contains 2000 instances. In this experiment we conduct an analysis of both MAR and MNAR missing data, in order to understand

Table 1: Accuracy of imputation for data simulated from different Bayesian networks with two MNAR missing values per variable. Avg. imput. acc. stands for Average imputation accuracy.

Bayesian net	Algorithm	Avg. imput. acc.
Breast Cancer	Exact learning	**84.38%**
	Approx. learning	**80%**
	Structural EM	50%
Prostate Cancer	Exact learning	**91%**
	Approx. learning	**86.36%**
	Structural EM	50%
ASIA	Exact learning	**84.38%**
	Approx. learning	**79%**
	Structural EM	43.75%

whether the benefits that we have seen before are only significant in the MNAR case. Thus, we carry out two experiments: (i) MNAR setting by randomly generating missing values all having the same data value (we repeat that to both zero and one values, one at a time); (ii) *MAR* setting by randomly generating missing values regardless of their respective original values. These simulations are repeated 100 times.

First, we generate two missing values per variable (24 missing values). A comparison between the imputation accuracy values of the approximate algorithm and structural EM is displayed in the first two rows of Table 2 (named 'Spread All Over'). Somewhat surprisingly, our new algorithm is significantly better than structural EM even when missing data are MAR.

Second, we generate 20 missing class values and repeat the experiment to span all instances such that each run involves missing values belonging to different instances (without replacement). For the MNAR experiment, each run consists of 20 identical missing class values (that is, we only make missing values of the same class, and we repeat that for both classes). For the MAR case, there is no such restriction and missing class values are randomly generated. Hence, there are 100 runs in order to cover all 2000 instances. Results of the approximate algorithm, structural EM and SVM using different kernels (for the sake of comparison with a state-of-the-art classifier) are displayed in the bottom rows of Table 2. Results of the proposed algorithm are significantly better when MNAR data are used, while the same cannot be stated for the MAR case.

SPECT Dataset

The Single Proton Emission Computed Tomography (SPECT) dataset consists of binary data denoting partial diagnosis from SPECT images (Lichman 2013). Each patient (data instance) is classified into one of two categories, normal and abnormal. The SPECT data consists of 267 instances and 23 variables in total (22 binary variables and a binary class variable). We generate MNAR missing data with different proportions, always using only one specific value (missing data proportions over all the data are 3%, 5% and 10%). These randomly generated datasets are given as input to the approximate algorithm as well as to structural EM. We

Table 2: Accuracy of imputation for experiments performed on the Lung Cancer dataset (LUCAS). *Spread All Over* refers to an imputation of 2 missing values per variable out of the 12 LUCAS variables. *Classification* refers to a classification problem performed as a cross-validation (100-fold cross-validation in the MNAR setting case) on LUCAS, using SVM, vs. an imputation task on the 20 missing class variables of the same folds, by both the proposed approximate learning algorithm and Structural EM. SVM kernels displayed are those that achieved the highest accuracy in each experiment. MP stands for missingness process, and rbf for radial basis function.

MP	Algorithm	Avg. imput. acc.
Exp.: Spread All Over		
MNAR	Approx. learning	**70.83%**
	Structural EM	45%
MAR	Approx. learning	70%
	Structural EM	50%
Exp.: Classification		
MNAR	Approx. learning	**97.5%**
	Structural EM	42.5%
	SVM (rbf)	45%
MAR	Approx. learning	69%
	Structural EM	70%
	SVM (rbf)	55%

note that there is a large discrepancy in the number of data values holding each of the two binary values: About 67% of the SPECT data has a value 0, whereas merely 33% of the data has a value 1. Due to that, we also investigate the average MNAR imputation accuracy within each data value separately, and note as well that there is some discrepancy in such accuracy values. Imputation accuracy of the approximate learning algorithm and structural EM are displayed in Table 3. The new algorithm is significantly better.

Smoking Cessation Study Dataset

The dataset used in this experiment is taken from a smoking cessation study as described in Gruder et al. (1993). It has been further utilized in other works, most notably Hedeker, Mermelstein, and Demirtas (2007). The smoking cessation dataset is a binary dataset consisting of 489 patient records (instances) with the missing data being inherently therein, i.e. there is no need to simulate missing data. The dataset contains 4 variables including the class variable, which refers to `smoking` or `non-smoking`. All the missing values are located in the class variable. There is a total of 372 patient records with observed classes, consisting of 294 smoking and 78 non-smoking records, as well as 117 records with missing class labels.

The experiment we perform here is a semi-supervised learning (SSL) experiment where we evaluate the performance of the algorithms as follows: (i) We hide the class labels of a portion of the observed labels; (ii) We apply the approximate learning on the data consisting of the originally missing and artificially hidden labels as missing values, and the rest of the data as observed values. Clearly this is a SSL

Table 3: MNAR imputation accuracy for the BN Approximate Learning algorithm and Structural EM on the SPECT dataset with various proportions of missing values, and for both data values. m.v. stands for missing value.

Missing values	Algorithm	Avg. imput. acc.
3% (overall)	New approx.	**81.75%**
	Structural EM	60%
5% (overall)	New approx.	**75.22%**
	Structural EM	49.27%
10% (overall)	New approx.	**81.94%**
	Structural EM	62.04%
3% (m.v. = 0)	New approx.	**95.65%**
	Structural EM	56.52%
5% (m.v. = 0)	New approx.	**80.43%**
	Structural EM	39.13%
10% (m.v. = 0)	New approx.	**92.75%**
	Structural EM	60.87%
3% (m.v. = 1)	New approx.	**67.83%**
	Structural EM	63.48%
5% (m.v. = 1)	New approx.	**70%**
	Structural EM	59.4%
10% (m.v. = 1)	New approx.	**71.13%**
	Structural EM	63.2%

experiment where the training data consists of the records with observed labels as labeled instances, records with originally missing labels as unlabeled instances, and the test instances are the records with artificially hidden labels.

The evaluation metric is the accuracy of the test instances using a cross-validation approach, as usually done in classification experiments. We compare the performance of the approximate algorithm against an equivalent procedure using structural EM (labels are then chosen based on the posterior distribution), and also against a semi-supervised learner in the form of a Laplacian SVM (Melacci and Belkin 2011) whose code is available online. Accuracies of the approximate algorithm, structural EM, and the semi-supervised Laplacian SVM are displayed in Table 4. Results suggest that the new algorithm is a very promising approach for SSL.

Table 4: MNAR Semi-supervised learning (SSL) results of the Smoking Cessation study data. All test records are Smoking records. The first column refers to the number of missing values in the test set. Accuracy expresses cross-validated accuracy of the test set.

# missing values	Algorithm	Avg. Accuracy
25	Approx. Learning	**90%**
	Structural EM	15%
	Laplacian SVM	76%
50	Approx. Learning	**88%**
	Structural EM	10%
	Laplacian SVM	73.5%
75	Approx. Learning	**88%**
	Structural EM	8%
	Laplacian SVM	76%

Car Evaluation Dataset

The Car Evaluation dataset (Blake and Merz 1998; Lichman 2013) contains 1728 instances and 7 variables consisting of 6 attributes and a class. The 6 attributes refer to the following: buying, maintenance, doors, persons, luggage boots and safety. The class variable refers to the car acceptability and can have exactly one of the following values: `unacceptable`, `acceptable`, `good`, `very good`. All variables are categorical with 3 or 4 states. The data were derived from a hierarchical decision model originally developed by Bohanec and Rajkovic (1988). Similar to the LUCAS experiment, a MNAR classification task is performed by involving missing values belonging all to one category of the class variable at a time (this is repeated for each label). Due to the class label unbalance (`unacceptable`: 1210 instances, `acceptable`: 384, `good`: 69, `v-good`: 65), we performed 10 experiments testing only the `unacceptable` and `acceptable` labels in five each, where there are 100 randomly chosen instances with a missing label (test set) in each experiment. The proposed algorithm is compared to structural EM and to an SVM classifier. Classification results are displayed in Table 5. Again, the new algorithm is significantly better than the others.

Table 5: Accuracy of classification for experiments performed on the Car Evaluation dataset. SVM with an rbf kernel is reported since it leads to best accuracy compared to other 5 experimented kernels.

Algorithm	Avg. Accuracy
Approximate Learning	**87.5%**
Structural EM	69.38%
SVM (rbf)	85.96%

Conclusions

In this paper we discuss the Bayesian network structure learning problem with missing data. We present an approach which performs well even when data are not missing at random. We define an optimization task to tackle the problem and propose a new exact algorithm for it which translates the task into a structure learning problem without missing data. Inspired by the exact procedure, we develop an approximate algorithm which employs structure optimization as a subcall. Experiments show the advantages of such approach. The proposed approximate method can scale to domains with hundreds or even thousands of variables. When the amount of missing data is exaggerated or, more interestingly, in cases where the existence of latent variables is a possibility, it will be intriguing to see how the proposed approximate method will fare. We intend to investigate such avenues in future work.

References

Almeida, E.; Ferreira, P.; Vinhoza, T.; Dutra, I.; Wu, Y.; and Burnside, E. 2014. ExpertBayes: Automatically refining manually built Bayesian networks. *Machine Learning and Applications (ICMLA)* 13:362–366.

Bartlett, M., and Cussens, J. 2013. Advances in Bayesian network learning using integer programming. *Conference on Uncertainty in Artificial Intelligence (UAI)* 29:182–191.

Blake, C., and Merz, C. 1998. UCI machine learning repository of machine learning databases.

Bohanec, M., and Rajkovic, V. 1988. Knowledge acquisition and explanation for multi-attribute decision making. *Intl. Workshop on Expert Systems and their Applications* 8:59–78.

Borchani, H.; Amor, N. B.; and Mellouli, K. 2006. Learning Bayesian network equivalence classes from incomplete data. *Lecture Notes in Comp. Science* 291–295.

Buntine, W. 1991. Theory refinement on Bayesian networks. *Conference on Uncertainty in Artificial Intelligence (UAI)* 7:52–60.

Chickering, D. 1996. Learning Bayesian networks is NP-complete. *Learning from Data* 121–130.

Cooper, G., and Herskovits, E. 1992. A Bayesian method for the induction of probabilistic networks from data. *Machine Learning* 9:309–347.

de Campos, C., and Ji, Q. 2011. Efficient structure learning of Bayesian networks using constraints. *Journal of Machine Learning Research (JMLR)* 12:663–689.

D'Orsi, C.; Bassett, L.; Berg, W.; Feig, S.; Jackson, V.; and Kopans, D. 2003. BI-RADS: Mammography. *American College of Radiology* 4.

Fogelman-Soulie, F. 2008. *Mining massive data sets for security: Advances in data mining, search, social networks and text mining, and their applications to security*, volume 19. IOS Press.

Friedman, N.; Geiger, D.; and Goldszmidt, M. 1997. Bayesian network classifiers. *Machine Learning* 29:131–163.

Friedman, N. 1998. The Bayesian structural EM algorithm. *Conference on Uncertainty in Artificial Intelligence (UAI)* 14:129–138.

Gruder, L.; Mermelstein, J.; Kirkendol, S.; D., D. H.; Wong, C.; Schreckengost, J.; Warnecke, R.; Burzette, R.; and Miller, T. 1993. Effects of social support and relapse prevention training as adjuncts to a televised smoking cessation intervention. *J Consult Clin Psychol* 61:113–120.

Hall, M.; Frank, E.; Holmes, G.; Pfahringer, B.; Reutemann, P.; and Witten, I. 2009. The WEKA data mining software: An update. *SIGKDD Explor. Newsl.* 11:10–18.

Heckerman, D.; Geiger, D.; and Chickering, D. 1995. Learning Bayesian networks: The combination of knowledge and statistical data. *Machine Learning* 20:197–243.

Hedeker, D.; Mermelstein, J.; and Demirtas, H. 2007. Analysis of binary outcomes with missing data: missing = smoking, last observation carried forward. *Addiction* 102:1564–1573.

Koivisto, M. 2006. Parent assignment is hard for the MDL, AIC, and NML costs. *conference on Learning Theory (COLT)* 19:289–303.

Lauritzen, S., and Speigelhalter, D. 1988. Local computations with probabilities on graphical structures and their application to expert systems. *Royal Statistical Society B* 50:157–224.

Leray, P., and Francois, O. 2005. Bayesian network structural learning and incomplete data. *Intl. and Interdisc. Conf. on Adaptive Knowledge Repr. and Reasoning (AKRR)* 33–40.

Lichman, M. 2013. UCI machine learning repository.

Little, R., and Rubin, D. 1987. *Statistical analysis with missing data*. John Wiley & Sons.

Meila, M., and Jordan, M. 1998. Estimating dependency structure as a hidden variable. *Advances in Neural Information Processing Systems (NIPS)* 584–590.

Melacci, S., and Belkin, M. 2011. Laplacian support vector machines trained in the primal. *Journal of Machine Learning Research (JMLR)* 12:1149–1184.

Mohan, K.; Pearl, J.; and Tian, J. 2013. Graphical models for inference with missing data. *Advances in Neural Information Processing Systems (NIPS)* 1277–1285.

Ramoni, M., and Sebastiani, P. 1997. Learning Bayesian networks from incomplete databases. *Conference on Uncertainty in Artificial Intelligence (UAI)* 13:401–408.

Rancoita, P.; Zaffalon, M.; Zucca, E.; Bertoni, F.; and de Campos, C. 2016. Bayesian network data imputation with application to survival tree analysis. *Computational Statistics & Data Analysis* 93:373–387.

Riggelsen, C., and Feelders, A. 2005. Learning Bayesian network models from incomplete data using importance sampling. *AISTATS* 301–308.

Riggelsen, C. 2006. Learning Bayesian networks from incomplete data: An efficient method for generating approximate predictive distributions. *SDM* 130–140.

Sarabando, A. 2011. Um estudo do comportamento de redes Bayesianas no prognóstico da sobrevivencia no cancro da prostata. *M.Sc. thesis, Universidade do Porto* 29:131–163.

Scanagatta, M.; de Campos, C.; Corani, G.; and Zaffalon, M. 2015. Learning Bayesian networks with thousands of variables. *Advances in Neural Information Processing Systems (NIPS)* 1855–1863.

Van den Broeck, G.; Mohan, K.; Choi, A.; and Pearl, J. 2014. Efficient algorithms for Bayesian network parameter learning from incomplete data. In *Causal Modeling and Machine Learning Workshop at ICML-2014*, R–441.