Multivariate imputation of qualitative missing data using Bayesian networks*

Vanessa Romero and Antonio Salmerón

Dept. Statistics and Applied Mathematics
University of Almería
La Cañada de San Urbano s/n
04120 Almería (Spain)
{avrofe,Antonio.Salmeron}@ual.es

Summary. In this paper we propose a methodology for the imputation of qualitative missing data using Bayesian networks. The idea is to learn a Bayesian network from the available complete data and use it to simultaneously impute all the missing cells in a register by means of abductive inference. The proposed methodology is experimentally tested and compared with the use of classification trees.

1 Introduction

Missing or incomplete data conform a serious problem that can be found very frequently in all kinds of surveys. The presence of missing data complicates the data analysis process, since most of the statistical procedures require complete samples. In many cases, the decision adopted by the analysts consists of discarding the registers with missing values and performing the analysis using only complete registers. The drawback of this approach is that the sample size can be significantly reduced and furthermore, much information contained in the sample can be lost; Namely, the information provided by the records with missing data.

When these records are discarded, it is assumed that the missingness mechanism is *completely at random* (MCAR), which means that it does not depend on the data (neither missing nor observed). A more realistic assumption is to consider that the missing data are *missing at random* (MAR), indicating that they depend on the observed values but not on the missing ones themselves. These issues are discussed in [13].

Most of the modern methods for missing data assume the MAR condition [16], and usually the missing values are imputed by first fitting a regression model to the observer variables that are correlated to the missing ones. When

*This work has been supported by the Spanish Ministry of Science and Technology through project TIC2001-2973-C05-02

the missing variables are qualitative, instead of regression models, in the last years classification trees have been successfully applied as imputation tools [1, 4].

Another commonly used approach is based on iterative algorithms that start from an initial rough model and impute the missing values using it. Afterwards, the model is improved taking into account the imputed values. The process is repeated until a steady state is reached. Examples of these methods are the EM algorithm [8], the methods based on Gibbs sampling [9] and the stochastic EM algorithm [3].

The rest of the paper is organised as follows. The problem of imputation of missing values is stated in section 2. We propose an iterative imputation algorithm in section 3. An incremental method is presented in section 4. The performance of the proposed method is experimentally tested in section 5 and the paper ends with a discussion of the results in section 6.

2 Imputation of missing values

Along this paper we assume a sample $\mathbf{S} = \{\mathbf{X}^{(1)}, \ldots, \mathbf{X}^{(n)}\}$ of qualitative random vectors (i.e. discrete random vectors with finite support). For each vector $\mathbf{X}^{(i)}$, some of its coordinates may be missing, and will be denoted by $\mathbf{X}_m^{(i)}$. The observed coordinates will be denoted by $\mathbf{X}_o^{(i)}$.

In general, the way in which imputation algorithms operate can be divided into two steps: (1) Learning a model from the data, and (2) imputing the missing values using the learnt model. These two steps can be repeated so that it is possible to start off with a rather easy model (for instance a uniform distribution) that is iteratively improved according to the observed data. For instance, the EM algorithm begins with an initial configuration of the parameters in the model and then imputes the missing values using the initial model. The parameters are then re-calculated as those that maximise the expected value of some sufficient statistic used to estimate them. The process is repeated until convergence is reached.

A similar behaviour is shown by the *data augmentation* (DA) algorithm, that is based on Gibbs sampling [9]. It starts from some initial assignment to the missing values and then iterates changing the values of the missing variables simulating from the distribution of each missing variable given the other ones (observed and missing). The simulation is repeated until an equilibrium is reached.

In the case of qualitative random vectors, in both algorithms it is fundamental the way in which the learnt model is represented. First of all, the number of parameters to estimate when learning a joint probability distribution for a set of qualitative variables grows exponentially with the number of variables. Furthermore, simulating from such a distribution can be a difficult task: It requires an initial configuration of the variables with positive probability and, besides, it is possible to reach a configuration that can be difficult

to change. These problems have been deeply studied within the context of Bayesian networks [10, 15].

3 Imputation using Bayesian networks

Bayesian networks have been successfully employed as efficient representations of joint distributions [6, 11]. A *Bayesian network* for a set of variables $\mathbf{X} = \{X_1, \ldots, X_k\}$ is an acyclic directed graph where each node represents a random variable and such that there is a conditional distribution $p(x_i|\mathbf{x}_{pa(i)})$ for each variable X_i given its parents in the graph, $\mathbf{X}_{pa(i)}$. A key property of Bayesian networks is that the joint distribution over $\mathbf{X}$ factorises as

$$p(\mathbf{x}) = \prod_{i=1}^{k} p(x_i|\mathbf{x}_{pa(i)}). \tag{1}$$

The fundamental of the algorithm we propose in this paper consists of using a Bayesian network as a model for the data, which is learnt from the observed variables $\mathbf{X}_o^{(i)}$, $i = 1, \ldots, n$. Once the model is obtained, a value for each $\mathbf{X}_m^{(i)}$, $i = 1, \ldots, n$ is simulated from it:

Algorithm IMPUTE_BN($\mathbf{S}, N$)

Input: A sample (database) $\mathbf{S} = \{\mathbf{X}^{(1)}, \ldots, \mathbf{X}^{(n)}\}$ with missing data, and the number of iterations (N).
Output: A database $\mathbf{S}'$ where the missing values have been imputed.

1. $\mathbf{S}_m := \{\mathbf{x} \in \mathbf{S} \mid \mathbf{x}$ contains missing values$\}$; $\mathbf{S}_o := \mathbf{S} \setminus \mathbf{S}_m$.
2. FOR $i := 1$ to N
   a) IF $(i = 1)$, $\mathbf{S}_l := \mathbf{S}_o$ ELSE $\mathbf{S}_l := \mathbf{S}_o \cup \mathbf{S}_m$.
   b) Learn a Bayesian network G from S_l.
   c) For each $\mathbf{x} = (\mathbf{x}_m, \mathbf{x}_o) \in \mathbf{S}_m$
      i. Generate a new value $\mathbf{x}_m$ for $\mathbf{X}_m$ from the distribution $p(\mathbf{x}_m|\mathbf{x}_o)$ computed from G.
      ii. Replace $\mathbf{x}_m$ in $\mathbf{x}$.
3. RETURN $\mathbf{S}' := \mathbf{S}_o \cup \mathbf{S}_m$.

Note that, in the first iteration, the network is learnt using only the complete registers in the database. In the next iterations, the imputed values are used as well. It is aimed at improving the quality of the database in each iteration. Another important feature of this algorithm is that the initial model (network) is not chosen at random, but trying to extract the best representation of the observed data. We describe now how the network is learnt (step 2b), as well as the generation of the values to impute (step 2(c)i).

3.1 Learning a Bayesian network from data

Several methods for learning Bayesian networks can be found in the literature (see [2] for instance). In this paper we have chosen the so-called K2 algorithm [5]. This method searches amongst the space of possible Bayesian networks for a given set of variables. The search starts with a network without arcs, and from then on, the network is completed inserting each time the arc that best increases its *quality*, as long as it does not make a cycle. The quality of the network is measured in terms of the likelihood of the data for such network. The algorithm ends when the inclusion of any new arc does not increase the quality of the network.

The reason to choose the K2 algorithm is that it is fast and produces accurate networks. However, any other algorithm can be employed.

3.2 Obtaining the values to impute from the learnt network

In step 2(c)i of algorithm **IMPUTE_BN**, a value $\mathbf{x}_m$ for the missing variables in a record is generated from the conditional distribution of the missing variables given the observed ones, $p(\mathbf{x}_m|\mathbf{x}_o)$. This distribution is not explicitly contained in the Bayesian network G, but can be computed from it. However, if the number of variables is high, dealing with that distribution may be infeasible, since the amount of probability values required to represent it grows exponentially with the number of variables.

We have approached this problem realising that, in practice, it may be unnecessary to manipulate all the probability values of $p(\mathbf{x}_m|\mathbf{x}_o)$. For instance, we could choose to impute the missing values using the mode of the distribution. The mode can be obtained from the Bayesian network using *abductive inference*, which consists of computing

$$\mathbf{x}_m^* = \arg\max_{\mathbf{x}_m} p(\mathbf{x}_m|\mathbf{x}_o) \ . \tag{2}$$

This value can be computed from the network without explicitly obtaining the joint distribution for $\mathbf{X}_m$ [14]. Nevertheless, imputing using the mode has some drawbacks. It is known that this technique does not preserve the distribution of the data, but rather increases its peaks. Besides, there are situations in which the mode has a probability value very close to other configurations of $\mathbf{X}_m$. For instance, if the mode has probability 0.201 and another configuration has probability 0.2, the last one would never be imputed. However, the difference between both probabilities could be due to estimation error.

One alternative approach could be to simulate a value $\mathbf{x}_m$ from $p(\mathbf{x}_m|\mathbf{x}_o)$, but again the problem of efficiency arises, since it would be necessary to handle all the values in the distribution. We have decided for a compromise between efficiency and precision, which consists of computing the M most probable values of $\mathbf{x}_m$ given $\mathbf{x}_o$ and then draw one of them according to their probabilities. The M configurations are obtained using Nilsson's algorithm

[14]. But still it is possible to find problems in which the computation of the M most probable configurations is not feasible. In order to deal with this situation, an approximate abduction algorithm can be used [7].

4 Incremental register imputation

In algorithm **IMPUTE_BN**, for each iteration a network is constructed and all the missing data are imputed from it. This approach can be refined if, at first, registers with few missing cells are completed, since they might contain valuable information to improve the quality of the learnt network. Then the network can be re-constructed and used to impute the rest of the missing data. The details of the algorithm incorporating this approach are as follows.

Algorithm INCR_IMPUTE_BN(S)

Input: A sample (database) $\mathbf{S} = \{\mathbf{X}^{(1)}, \ldots, \mathbf{X}^{(n)}\}$ with missing data.
Output: A database $\mathbf{S}'$ where the missing values have been imputed.

1. Let k be the number of variables in each item of the sample.
2. $\mathbf{S}_m := \{\mathbf{x} \in \mathbf{S} \mid \mathbf{x}$ contains missing values$\}$; $\mathbf{S}_o := \mathbf{S} \setminus \mathbf{S}_m$.
3. FOR $i := 1$ to k
   a) $\mathbf{S}_i := \{\mathbf{x} \in \mathbf{S}_m \mid \mathbf{x}$ has i missing values$\}$.
   b) $\mathbf{S}_m := \mathbf{S}_m \setminus \mathbf{S}_i$.
   c) Learn a Bayesian network G from S_o.
   d) For each $\mathbf{x} = (\mathbf{x}_m, \mathbf{x}_o) \in \mathbf{S}_i$
      i. Generate a new value $\mathbf{x}_m$ for $\mathbf{X}_m$ from the distribution $p(\mathbf{x}_m|\mathbf{x}_o)$ computed from G.
      ii. Replace $\mathbf{x}_m$ in $\mathbf{x}$.
   e) $\mathbf{S}_o := \mathbf{S}_o \cup \mathbf{S}_i$.
4. RETURN $\mathbf{S}' := \mathbf{S}_o$.

Observe that this algorithm, unlike the former one, is incremental rather than iterative. Of course, the process could be repeated but the trials we have carried out over several examples suggest that the effort invested is not worth.

5 Experimental evaluation

We have carried out a series of experiments in order to test the performance of the algorithms proposed in this paper. We have considered three databases:

- `chest`: A database with 8 variables and 2000 registers generated using forward sampling from the chest clinic network used in [12].

- `water`: A 32-variable database with 4000 registers generated by forward sampling from the `water` network borrowed from the Decision Support Systems group at Aalborg University (`http://www.cs.auc.dk/research/DSS/Misc/networks.html`).
- `housing`: A database with 13 variables and 506 registers taken from the UCI Machine Learning Repository (`http://www.ics.uci.edu/~mlearn/MLRepository.html`).

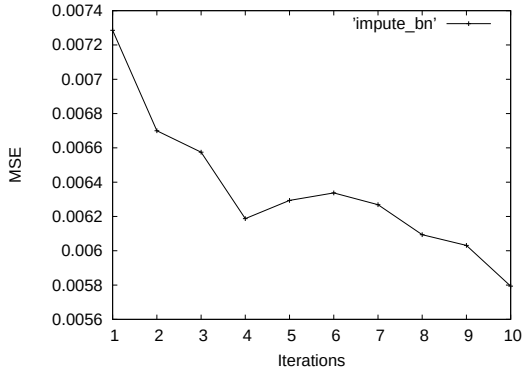

Fig. 1. Results of algorithm **IMPUTE_BN** for database `chest` with 10 iterations

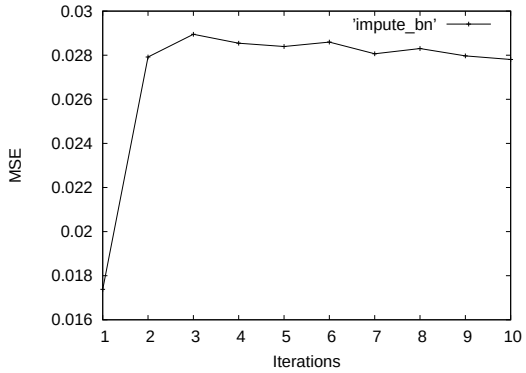

Fig. 2. Results of algorithm **IMPUTE_BN** for database `housing` with 10 iterations

In each database, a 60% of their registers have been selected at random and, in the selected registers, each cell is set as missing with probability equal to 0.4.

We have tested algorithms **IMPUTE_BN** and **INCR_IMPUTE_BN**, and compared their performance with an algorithm based on classification trees.

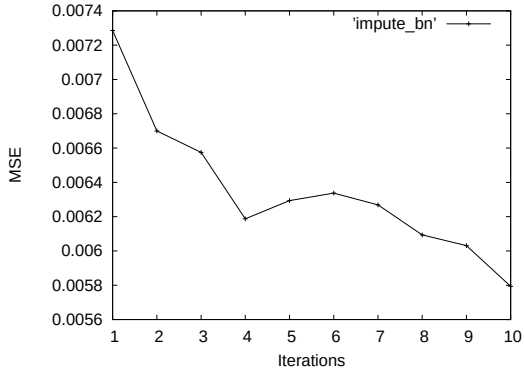

Fig. 3. Results of algorithm **IMPUTE_BN** for database `water` with 10 iterations

Table 1. Comparative results of imputation using classification trees, algorithm **INCR_IMPUTE_BN** and **IMPUTE_BN** after 10 iterations

	Classification tree	**INCR_IMPUTE_BN**	**IMPUTE_BN**
`chest`	0.0328	0.0051	0.0058
`housing`	0.0371	0.0157	0.0278
`water`	0.0328	0.0051	0.0058

All these methods have been implemented within the Elvira system, which is available from `http://leo.ugr.es/~elvira`.

The accuracy of the imputation is measured in terms of the mean squared error (MSE) of the marginal distribution of each imputed variable (computed from the imputed database) with respect to its exact distribution (computed from the original database). The results of algorithm **IMPUTE_BN** for 10 iterations are displayed in figures 1, 2 and 3, while in table 1 the proposed algorithms are compared with the use of classification trees.

6 Discussion

According to the experimental results, the proposed algorithms outperform the one based on classification trees, which produces very good results as well. It was expectable since Bayesian networks are known to be more a powerful tool for classification than trees. With respect to the two proposed methods, incremental imputation seems to be the best choice rather than the iterative algorithm.

The reason why the results do not improve very much as the number of iteration increases (even in the experiment with the `housing` database the error increases in the second iteration) is that the initial model is very

accurate, due to the refinement of existing learning procedures for Bayesian networks.

As future works, we plan to extend the methods to continuous variables and study their performance in presence of functional dependencies, where the representation power of Bayesian networks should make a difference.

References

1. M.J. Bárcena and F. Tusell. Multivariate data imputation using trees. Technical Report BILTOKI # 200205, Dept. Estadística y Econometría. Universidad del País Vasco, 2002.
2. E. Castillo, J.M. Gutiérrez, and A.S. Hadi. *Expert systems and probabilistic network models*. Springer-Verlag, New York, 1997.
3. G. Celeux and J. Diebolt. The SEM algorithm: A probabilistic teacher algorithm derived from the EM algorithm for the mixture problem. *Computational Statistics Quarterly*, 2:73–82, 1985.
4. C. Conversano and C. Cappelli. Missing data incremental imputation through tree based methods. In *Proceedings of the COMPSTAT 2002 Conference*, pages 455–460. Springer Verlag, 2002.
5. G.F. Cooper and E. Herskovits. A Bayesian method for the induction of probabilistic networks from data. *Machine Learning*, 9:309–347, 1992.
6. R.G. Cowel, A.P. Dawid, S.L. Lauritzen, and D.J. Spiegelhalter. *Probabilistic networks and decision graphs*. Springer, 1999.
7. L.M. de Campos, J.A. Gámez, and S. Moral. Partial abductive inference in Bayesian networks by using probability trees. In *Proceedings of the 5th International Conference on Enterprise Information Systems (ICEIS'03)*, pages 83–91, Angers, 2003.
8. A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximun likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society B*, 39:1 – 38, 1977.
9. W.R. Gilks, S. Richardson, and D.J. Spiegelhalter. *Markov chain Monte Carlo in practice*. Chapman and Hall, London, UK, 1996.
10. C.S. Jensen, A. Kong, and U. Kjærulff. Blocking Gibbs sampling in very large probabilistic expert systems. *International Journal of Human-Computer Studies*, 42:647–666, 1995.
11. Finn V. Jensen. *Bayesian networks and decision graphs*. Springer, 2001.
12. S.L. Lauritzen and D.J. Spiegelhalter. Local computations with probabilities on graphical structures and their application to expert systems. *Journal of the Royal Statistical Society, Series B*, 50:157–224, 1988.
13. R.J.A. Little and D.B. Rubin. *Statistical analysis with missing data*. John Wiley and Sons, New York, 1987.
14. D. Nilsson. An efficient algorithm for finding the M most probable configurations in Bayesian networks. *Statistics and Computing*, 9:159–173, 1998.
15. A. Salmerón, A. Cano, and S. Moral. Importance sampling in Bayesian networks using probability trees. *Computational Statistics and Data Analysis*, 34:387–413, 2000.
16. J.L. Schafer. *Analysis of incomplete multivariate data*. Chapman and Hall, 1997.