Tree-based Algorithms for Missing Data Imputation

M.J. Bárcena[1] and F. Tusell[1]

[1] Facultad de CC.EE. y Empresariales, Universidad del País Vasco / Euskal Herriko Unibertsitatea, Avda. Lehendakari Aguirre, 83, 48015 BILBAO, Spain. Research supported through grant PB98-0149 from CICYT. Comments by the editors gratefully acknowledged.

Abstract. Let $\mathbf{X}$ be a $N \times (p+q)$ data matrix, with entries partly missing in the last q columns. A problem of practical relevance is that of drawing inferences from such an incomplete data set. We propose to use a sequence of trees to impute missing values. Essentially, the two algorithms we introduce can be viewed as predictive matching methods. Among their advantages, their flexibility, which makes no assumptions about the type or distribution of the variables.

Keywords. missing data; multiple imputation; binary trees; file matching; predictive matching.

1 Introduction

Let $\mathbf{X}$ be a $N \times (p+q)$ data matrix, with entries partly missing in the last q columns. This includes: i) Scattered missing entries which were never recorded, or were subsequently lost; ii) A full block missing, say the $n_{\text{mis}} \times q$ block consisting of the last n_{mis} rows and q columns. The second situation may arise, for instance, if $\mathbf{X}$ contains data collected in two surveys given to n_{obs} and n_{mis} subjects respectively ($N = n_{\text{obs}} + n_{\text{mis}}$), and complete data is available for the n_{obs} subjects interviewed in the first survey while the last q questions were not asked to the n_{mis} subjects interviewed in the second survey. Only case ii) is dealt with in this paper, although generalization is possible.

A problem of practical relevance is that of drawing inferences from such an incomplete data set, and a considerable body of literature exist on this issue. A landmark is the monograph Little and Rubin(1987), setting up a methodology and advocating the use of multiple imputation. A recent monograph is Schafer(1997) which develops algorithms for imputation based on the EM algorithm and data augmentation.

2 Goals

The methods in Schafer(1997) require the specification of a parametric model and a (possibly non-informative) prior on the parameters. Our intent has been to produce a good all-purpose nonparametric method, capable of coping with situations where little is known about the underlying data generation mechanism.

Our research was initially motivated by the problem of completing a partially observed sample with regular structure (problem ii) in the Introduction). For instance, we might have a file with $N = n_{\text{obs}} + n_{\text{mis}}$ subjects. The

first n_{obs} subjects have been totally observed on $(p+q)$ variables. For the remaining n_{mis} subjects, only the first p variables have been observed. We would like to impute the missing q variables on these n_{mis} subjects with a method:

1. Making as little assumptions as feasible on the joint distribution of the $(p+q)$ variables;
2. Allowing for multiple imputation, and
3. Taking into account the structure of the q variables that are imputed.

This last point was of particular interest to us. We had an application where the q variables to impute where the times devoted by each subject to different tasks, and were required to add up to twenty four hours (see Bárcena and Tusell(1998,1999)). It was clear that those q variables had to be jointly imputed to ensure mutual compatibility.

Binary trees are a flexible tool to capture the relationship between a response and a set of predictors. However, neither in the seminal work Breiman et al.(1984) nor in the large body of literature that followed could we find examples in which the response was multivariate[1] and jointly imputed.

Next section describes two algorithms built around univariate response binary trees and designed to meet the three goals above.

3 Algorithms

We use a collection of ordinary (scalar response) binary trees. They are built with the methodology described by Breiman et al.(1984) as implemented by Therneau and Atkinson(1997); but a different strategy can be used (e.g., Murthy et al.(1994)). We denote by $\mathcal{Y}_{x|\,\mathbf{z}}$ a tree "regressing" x on the predictors in $\mathbf{z}$ —the response x might just as well be qualitative, and the tree a classification rather than a regression tree. Assume we have a training sample of n_{obs} subjects, fully observed in $(p+q)$ variables, while for the remaining n_{mis} subjects we only observe the first p variables. Call $\mathbf{X}_{\text{obs}}$ the vector of the p variables fully observed (for all $n_{\text{obs}}+n_{\text{mis}}$ subjects) and $\mathbf{X}_{\text{mis}}$ the vector of the q variables incompletely observed (missing for the last n_{mis} subjects). The case where observations are missing irregularly can also be handled (using surrogate splits), but we will only deal with case ii) of the Introduction in the following. We propose the following imputation strategies.

3.1 The forest climbing algorithm

It can be summarized as follows:

1. Build trees $\mathcal{Y}_{X_{p+1}|\,\mathbf{X}_{\text{obs}}},\ldots,\mathcal{Y}_{X_{p+q}|\,\mathbf{X}_{\text{obs}}}$ using the CART methodology and the n_{obs} complete observations.
2. Drop each of the n_{mis} incomplete cases down the q trees constructed. Let case i fall in the terminal nodes labelled $(\ell_{i,1},\ldots,\ell_{i,q})$ of (respectively) trees $\mathcal{Y}_{X_{p+1}|\,\mathbf{X}_{\text{obs}}},\ldots,\mathcal{Y}_{X_{p+q}|\,\mathbf{X}_{\text{obs}}}$. Call $(\ell_{i,1}\cap\ldots\cap\ell_{i,q})$ the subset of the n_{obs} complete cases which also end in said leaves. If $(\ell_{i,1}\cap\ldots\cap\ell_{i,q})\neq\emptyset$, impute the missing values of case i by those of one complete case which also ends in $(\ell_{i,1}\cap\ldots\cap\ell_{i,q})$. If multiple imputation is desired, sample k cases out of that intersection.

[1] Note the work Ciampi(1991): it does require the specification of a likelihood, though. Recently, Siciliano and Mola(2000) address the problem of constructing trees with multivariate response in a non-parametric way.

3. If $(\ell_{i,1} \cap \ldots \cap \ell_{i,q}) = \emptyset$, iteratively replace leaves by their ancestors ("climb the trees") until a non empty intersection is found from which one or more complete cases can be drawn.

The idea is disarmingly simple. Take any tree $\mathcal{Y}_{X_k|\,\mathbf{X}_{\text{obs}}}$, $p < k \leq p+q$. The leaves of that tree are classes of a partition of the predictor space such that, within each class, knowledge of $\mathbf{X}_{\text{obs}}$ cannot help us in further refining our prediction of X_k (otherwise, the leave would have been splited). It then makes sense that if subject i with unknown X_k ends in leave $\ell_{i,k}$ when dropped down the tree $\mathcal{Y}_{X_k|\,\mathbf{X}_{\text{obs}}}$, its X_k value be predicted by a function of the X_k values of subjects in the training sample which ended in the same leave. This function can be the mean, median or other summary statistic; or else we can sample from that leave if multiple imputation is desired.

Since we want to jointly impute all values in $\mathbf{X}_{\text{mis}}$ for the subject at hand, we would like to use complete cases in $(\ell_{i,1} \cap \ldots \cap \ell_{i,q})$, and this is exactly what the algorithm above does. The only additional caveat is that the relevant intersection might be empty —not one of the subjects in the training sample ended in exactly the same leaves than the subject to impute. If that is the case, the algorithm replaces nodes by their ancestors ("climbs the trees"), until a non empty intersection is found. The order of climbing is governed by the deviance —we climb first the tree where the replacement of a node by its ancestor leads to the least possible increase in deviance. Inasmuch as the deviance is scale-dependent, this is only an *ad-hoc* device.

We can think of the forest climbing algorithm as a nearest neighbour method in which "nearness" is defined as "falling in the same leaves than". Similar ideas exist in the literature, under the name of predictive mean matching.

3.2 The cascade algorithm

The cascade algorithm is directed at finding subjects in the training sample that are simultaneously "close" to the subject to impute in the metrics defined by all trees, obviating the need to climb.

Again the idea is quite simple. Jointly imputing $\mathbf{X}_{\text{mis}}$ given the values in $\mathbf{X}_{\text{obs}}$ is easy as soon as we have the conditional distribution $f(\mathbf{X}_{\text{mis}}|\mathbf{X}_{\text{obs}})$: we only have to draw from that distribution to perform multiple imputation. By successively conditioning, we can write

$$\begin{aligned} f(\mathbf{X}_{\text{mis}}|\mathbf{X}_{\text{obs}}) = \quad & f(X_{p+q}|\mathbf{X}_{\text{obs}}, X_{p+1}, \ldots, X_{p+q-1}) \\ & \times f(X_{p+q-1}|\mathbf{X}_{\text{obs}}, X_{p+1}, \ldots, X_{p+q-2}) \\ & \times \cdots \\ & \times f(X_{p+1}|\mathbf{X}_{\text{obs}}) \end{aligned}$$

We can regard a tree as a mechanism generating observations with a given conditional distribution. For instance, if we construct the tree $\mathcal{Y}_{x|\,\mathbf{X}_{\text{obs}}}$ we can generate approximate random drawings from $f(X|\mathbf{X}_{\text{obs}})$ by dropping $\mathbf{X}_{\text{obs}}$ down $\mathcal{Y}_{x|\,\mathbf{X}_{\text{obs}}}$ and sampling from the leave where it ends.

To generate observations with approximate distribution $f(\mathbf{X}_{\text{mis}}|\mathbf{X}_{\text{obs}})$ we can do the following:

1. Construct trees $\mathcal{Y}_{X_{p+1}|\,\mathbf{X}_{\text{obs}}}, \mathcal{Y}_{X_{p+2}|\,\mathbf{X}_{\text{obs}}X_{p+1}}, \ldots, \mathcal{Y}_{X_{p+q}|\,\mathbf{X}_{\text{obs}}X_{p+1},\ldots,X_{p+q-1}}$.
2. For each incomplete observation with observed $\mathbf{X}_{\text{obs}}$,

(a) Drop $\mathbf{X}_{\text{obs}}$ down the first tree. Sample the leave where it ends to obtain a value X_{p+1}.
(b) For $j = 2, \ldots, q$ do likewise: drop $\mathbf{X}_{\text{obs}}, \ldots, X_{p+j-1}$ down the j-th tree and sample the leave where it ends to obtain a vector of imputed values $X_{p+1}, \ldots, X_{p+j}$.

Note that while a joint distribution can be factored in any order, in the tree cascade algorithm just sketched order does matter. The ideal would be to reorder variables $X_{p+1}, \ldots, X_{p+q}$ in such a way that we have first those which can be best predicted from $\mathbf{X}_{\text{obs}}$ and last those which cannot be predicted well from $\mathbf{X}_{\text{obs}}$ yet are closely related to previously predicted variables. These are potentially conflicting criteria, and there is no clear choice. We have investigated two different alternatives: *best first* and *best last*. In the first case, the trees are used in order of decreasing goodness of fit; the rationale being that, since each imputed variable can be input in subsequent trees, we want the values imputed earlier to be of the best possible quality.

On the other hand, in order to ensure consistency of the imputed variables, the whole vector $X_{p+1}, \ldots, X_{p+q}$ is imputed at the last step, which makes desirable a high quality tree at the end of the cascade.

4 Implementation and simulated results

We have written functions to implement our methods in the statistical language R (see Venables et al.(1997) for a description). We have used the functions in the `rpart` package (see Therneau and Atkinson(1997)) as building blocks. For the purpose of comparison, we used the routines in the package `norm`, a port[2] to R of the programs of the same name described in Schafer(1997).

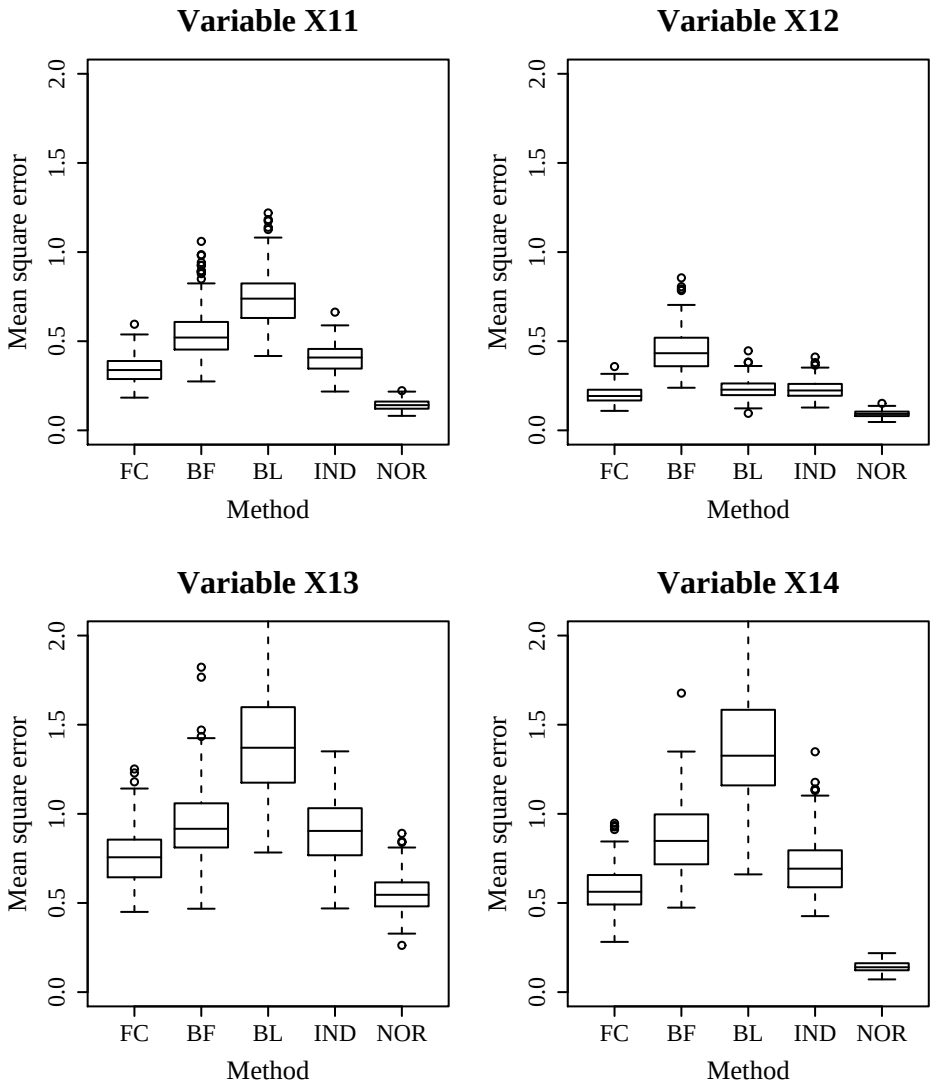
We have generated data from a multivariate normal distribution $N_{15}(\mathbf{0}, \Sigma)$ with Σ exhibiting moderate correlation among variables. The variables were standardized to have variance equal to one. Each of the two hundred replications generated contains $N = 500$ observations. The last $n_{\text{mis}} = 50$ observations of the last $q = 5$ variables were deleted and then their values imputed using the remaining $n_{\text{obs}} = 450$ complete observations as the training sample.

We have simulated the behaviour of the forest climbing algorithm (FC) and the cascade algorithm, both with best first (BF) and best last (BL) orderings and joint imputation (that is, all of the missing values are imputed at once, thus ensuring compatibility of the imputed values). We have also simulated the behaviour of the cascade algorithm with BF order and individual imputation of each variable (IND). Finally, we have simulated Schafer's method (NOR), using the EM algorithm to find the maximum likelihood estimates of the parameters conditional on $\mathbf{X}_{\text{obs}}$ and subsequently drawing random observations from that conditional distribution $f(\mathbf{X}_{\text{mis}}|\mathbf{X}_{\text{obs}})$.

Figure 1 shows the mean square error (MSE) of imputation for four of the variables, averaged over the 200 replications of the experiment (the fifth variable, not shown for lack of space, behaves similarly). Notice that a naïve strategy of imputing with a random complete subject from the sample *(cold deck)* would give a MSE of 2. Naïve imputation using the mean would give a MSE of 1. Since data is generated following a multivariate normal model, we can expect the parametric method (NOR) to perform best, and this is indeed the case. What is interesting is that the forest climbing algorithm is always a

[2] By Alvaro Novo, and available at CRAN, `http://cran.ar.r-project.org`.

Fig. 1. Imputation results for four variables and two hundred replications with $n_{\text{obs}} = 450$, $n_{\text{mis}} = 50$, $p = 10$, $q = 5$ and multivariate normally distributed data. See text for description of the methods.



second best. When imputing using the cascade algorithm, the minimum MSE is of course obtained imputing each variable separately. Of the remaining two cascade algorithms, neither order BF or BL seems uniformly better (see for example the results for variable X11 and X12 in Figure 1). Additional more extensive results are available from the authors.

5 Some remarks

As mentioned previously, both the forest climbing and cascade algorithms can be seen as *ad hoc* methods of predictive matching: they replace in block the missing values of a subject with those of another subject in the training sample that is close. "Close" is taken to mean that both would have similar predicted values when dropped down the set of trees constructed. It is important to notice that this notion of closeness is ambiguous, because we are jointly imputing $X_{p+1}, \ldots, X_{p+q}$. If the scales vary widely and/or there is strong correlation, it makes sense to rescale the variables and/or transform them to principal components before using the intersection method. The cascade method explicitly takes into account the relation among the responses: the ambiguity resurfaces in the ordering of the trees in the cascade.

Both methods scale well, and can be used with fairly large samples. The largest portion of time is devoted to constructing the trees. Subsequent imputation is very fast. Typically, only a fraction of cases require climbing in the forest climbing algorithm: in an application with a training sample of 2521 subjects $p = 5$ predictors and $q = 24$ variables to impute, under 2% of the subjects imputed required climbing. Once the q trees needed have been constructed, the (time) complexity of the algorithms is $O(qn_{\text{mis}})$, i.e. linear in the product of variables to impute times the number of cases to impute.

Both algorithms presented meet the goals enumerated in Section 2: they are all-around methods making almost no assumptions, take into account the structure of the variables to impute and provide for easy multiple imputation. We remark in closing that generalizations are possible to the case of irregularly missing observations.

References

M.J. Bárcena and F. Tusell. (1998). Linking surveys using reciprocal classification trees. In K. Fernández-Aguirre and A. Morineau (eds.) *Analyses Multidimensionnelles des Donnes*, Cisia-Ceresta, Saint-Mandé, 133–148.

M.J. Bárcena and F. Tusell. (1999). Enlace de encuestas: una propuesta metodológica y aplicación a la Encuesta de Presupuestos de Tiempo. *Qüestiió*, **23**, 297–320.

L. Breiman, J.H. Friedman, R.A. Olshen, and C.J. Stone. (1984). *Classification and Regression Trees.* Belmont, CA.: Wadsworth.

A. Ciampi. (1991). Generalized regression trees. *Computational Statistics and Data Analysis*, **12**, 57–78.

R.J.A. Little and D.B. Rubin. (1987). *Statistical Analysis with Missing Data.* New York: John Wiley and Sons.

S.K. Murthy, S. Kasif, and S. Salzberg. (1994) A system for induction of oblique decision trees. *Journal of Artificial Intelligence Research*, **2**, 1–32.

J.L. Schafer. (1997). *Analysis of Incomplete Multivariate Data.* London: Chapman and Hall.

R. Siciliano and F. Mola. (2000). Multivariate data analysis and modeling thorugh classification and regression trees. *Computational Statistics and Data Analysis*, **32**, 285–301.

T.M. Therneau and E.J. Atkinson. (1997). An introduction to recursive partitioning using the RPART routines. Technical Report, Mayo Foundation.

B. Venables, D. Smith, R. Gentleman, and R. Ihaka. (1997). *Notes on R: A Programming Environment for Data Analysis and Graphics.* Auckland: Dept. of Statistics, University of Adelaide and University of Auckland. Available at `http://cran.at.r-project.org/doc/R-intro.pdf`.