Multivariate Data Imputation using Trees

María Jesús BARCENA* Fernando TUSELL*

Abstract

We address the problem of completing two files with records containing a fully observed common subset of variables. The technique investigated involves the use of regression and/or classification trees. An extension of current methodology (the intersection-seeking or "forest-climbing" algorithm) is proposed to deal with multivariate response variables. The method is demonstrated and shown to be feasible and have some desirable properties.

Keywords: file completion; imputation; regression trees.

Acknowledgements. We thank for support the Spanish MEC (grant PB98-0149). We gratefully acknowledge comments from Vicente Nuñez, Eva Ferreira, Karmele Fernández, and participants of the IWSM'15 and COMPSTAT'2000 meetings. Any errors and obscurities that remain are our own.

1 Introduction

1.1 The problem

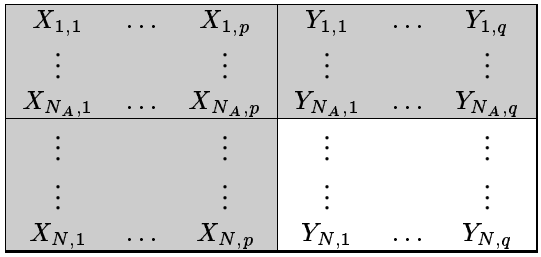
Our starting point are two files, A and B with a total of $N = N_A + N_B$ observations. When stacked, they can be seen as a single table with the structure shown in Figure 1. The shaded area corresponds to observed values and the unshaded area to missing ones.

We deal with the problem of imputing the missing values in the file B, using the values of the common variables $X_1, \ldots, X_p$. In other words, we address the issue of imputing the non shaded areas.

In the problem that motivates this research, this two files contain data from two sample surveys which share a common set of questions. Variables $X_1, \ldots, X_p$ are thus known for all N cases, but variables $Y_1, \ldots, Y_q$ have only been collected for the N_A cases in the first survey. In general, we may have more than two surveys and patterns of missingness more complicated than the one in Figure 1; but the simplified setting adopted suffices for the purposes of our exposition.

*Departamento de Estadística y Econometría. Facultad de CC.EE. y Empresariales, Avda. del Lehendakari Aguirre, 83, 48015 BILBAO. E-mail: etptupaf@bs.ehu.es.

Figure 1: Structure of the problem. The unshaded area of the table is missing.



1.2 Outline of the paper

Section 2 describes very briefly some of the techniques that have been used for survey imputation or file matching, and provides some pointers to the literature. Section 3 introduce the proposed method, against the background of the existing techniques. Section 4 contains results of simulations showing the performance of the proposed method. Some concluding remarks in Section 5 close the paper.

2 Imputation techniques

A short description of imputation techniques and some pointers to the literature are given next. The interested reader may also refer to Nordholt (1998).

2.1 Regression and the EM algorithm

A quite natural idea is to use the cases with complete values of variables (X, Y) to fit regressions of Y on X and then use these regressions to impute the missing values by $\hat{Y}$. This idea goes back at least to 1960, (see Buck (1960) for an early proponent).

The implied assumption when using regressions in this manner is the constancy of the relationships among the predictors X and the responses in both surveys. Obviously, a good fit of the regression models is also required, if the imputation is to be any good.

It is important to realize that *even if the above assumption is justified,* the imputed values will lack the variability of the genuine values: we are replacing unknown values *about* the regression hyperplane by imputed values *on* the hyperplane. This has to be corrected or taken into account in any subsequent analysis.

The EM algorithm advocated in Dempster et al. (1976) (see also Rubin (1991)) provides an easy, iterative way to maximize the likelihood function of a model with incomplete data in a wide variety of settings. We can use the model with maximum likelihood estimates replacing the parameters to generate imputations.

2.2 Nearest neighbours and deck replacement

Another common idea is to replace the missing values in one case by those of another case in some sense "close" to it, according to a predefined notion of "closeness" in the space of common variables X. For instance, to impute $\mathbf{Y_i} = (Y_{i,1}, \ldots, Y_{i,q})^T$ for $i \in \{N_A + 1, \ldots, N\}$ we could use $\hat{\mathbf{Y}}_\mathbf{i} = \mathbf{Y_j}$ for some $j \in \{1, \ldots, N_A\}$ such

that $\mathbf{X_i} \simeq \mathbf{X_j}$. This gives rise to a variety of flavours of the nearest neighbour idea, depending on how we define proximity in the space $\mathcal{X}$ of the X variables.

Sometimes, "closeness" means "close in the card deck", reflecting the practice of replacing the missing values in one case with those of the case next to it in the computer card deck —a reasonable procedure if the order in the deck reflects geographical contiguity or otherwise similarity among cases. See for instance (Rubin, 1987, p. 60).

In spite of their simplicity, deck replacement methods have advantages: the imputed values do not suffer from the lack of variability that afflict the regression imputed values. Also, the imputed values belong to some other case in the sample, hence are realistic and internally coherent. We will turn to this issue later.

Deck replacement methods are critically dependent on the rules used to determine the nearest neighbours. When this rules are poorly chosen, the quality of the imputations may suffer greatly.

2.3 Neural networks

Artificial Neural Networks have shown great usefulness in many problems, as universal approximators. They are ideally suited to model complex relationships when there is no clear choice of a parsimonious model. Useful monographs are Ripley (1996) and Bishop (1996). Nordbotten (1996) and Villagarcía and Muñoz (1997) are examples of uses in survey imputation.

2.4 Multiple imputation

Not an imputation technique, but rather a methodology that can improve many of them: Rubin (1987) and Little and Rubin (1987) convincingly show its rationale and benefits. See also Rubin (1986). Basically, the idea is to construct the distribution of the missing values conditional on the observed ones (the predictive distribution) and *sample* from this distribution. Rather than replacing each missing value with a single imputation (such as the conditional mean), we draw from the predictive distribution several such replacements *(proper imputations)*.

The idea is to generate not one but several complete data sets. In our setting, we would create several matrices such as the one in Figure 1, sharing the shaded areas but with different imputations for the missing data. We can then perform several classical, complete data analysis, and compare them to have an idea on how much the results vary due to random fluctuation in the imputation.

Of course the problem lies in obtaining a suitable approximation to the predictive distribution, and considerable effort has been expended in this area in the last years. Schafer (1997) describes how to create multiple imputations under a variety of multivariate models. The techniques rely heavily on Markov Chain Monte Carlo (MCMC) to generate approximate drawings from the predictive distributions.

A different strategy is adopted in the chained equations approach (see van Buuren and Oudshoorn (1999), Oudshoorn et al. (1999), van Buuren and Oudshoorn (2000)): univariate models are fitted to each univariate response to approximate the conditional distribution given all other variables. Then, the Gibbs sampler is used to derive approximate drawings from the joint multivariate distribution.

2.5 Imputation using regression or classification trees

We propose the use of regression and/or classification trees to impute missing values. Using trees has a number of advantages: it gives a unified treatment of continuous and categorical variables, provides useful byproducts to assess the goodness of fit and makes multiple imputation easy. Trees also have well known advantages: flexibility, few assumptions, relative insensitivity to outliers, etc. The seminal book Breiman et al. (1984) describes these advantages.

Consider the simplest possible case in which we have p common variables X and $q = 1$, i.e. there is only one specific variable to impute (refer to Figure 1, p. 2). The case (usually more relevant in practice) $q > 1$, is taken up in Section 2.6, and a method is proposed in Section 3.

Let $\mathcal{X}$ be the space of all possible values of X. A tree of Y on the X induces a partition of $\mathcal{X}$ such that in each class we have like values of Y. Since no restrictions are imposed on the kind and distributions of the variables, the CART methodology described in Breiman et al. (1984) seems a good way to build such partition. To impute a case, we drop it down tree and look at the leave where it ends. This is formalized in Algorithm 1, pág. 4.

Algorithm 1 – Univariate imputation using trees.

---

1. Build a tree $\mathcal{Y}_X$ "regressing" Y on the X, using cross validation and observations $i = 1, \ldots, N_A$. Let $\mathcal{Y}_1, \ldots, \mathcal{Y}_a$ the leaves of said tree, and $\mathcal{Y}$ the partition they form.

2. To impute the value of Y for a case with $i \in \{N_A + 1, \ldots, N\}$, drop it down the tree $\mathcal{Y}_X$. If it falls in the leave $\mathcal{Y}_{\delta(i)}$, impute Y as a function of the values of Y observed in that leave.

Notice that the method described lends itself quite well to something close to multiple imputation, since each case will normally end in a terminal node which contains cases with more than one value of Y. Thus, we can sample among them at random to create multiple imputations, if we consider the tree as giving a sufficiently good approximation to the predictive distribution. We can also impute using the mean, median, mode, etc., if we want a single imputation.

It is worth mentioning that Algorithm 1 can be seen as a regression method — only a tree replaces familiar linear or nonlinear regression, with both advantages and disadvantages. It can also be seen as a nearest neighbour method. But, while a nearest neighbour method using, for instance, Mahalanobis distance in the space $\mathcal{X}$, would disregard the values of the Y, here a different notion of proximity is used. A case is "near" another if both happen to fall in the same leave when dropped down the relevant tree. Thus, the notion of proximity used does take into account the response variable: the method is an instance of predictive matching. This is quite important and further discused later.

The use of trees to impute univariate missing values is already common practice and an option for imputation in several statistical and data mining packages.

2.6 Trees for multivariate imputation

When we attempt to generalize the method to multivariate Y (i.e., q>1), we stumble upon a pitfall. We would like a method to construct trees partitioning the $\mathcal{X}$ space in such a way that each class contains like values of the (multivariate) response: but there is no unique way to define likeness in a multidimensional space. A possibility is to use Kullback-Leibler or a similar measure of discrepancy as in Ciampi (1991), but this requires a model for the distribution of the response variables. Another posibility when all responses are categorical is to create a new response with $D = \prod_{j=1}^{q} d_j$ levels, where d_j is the number of levels in the j-th response and q the number of responses. This approach is advocated, for example, in Mesa et al. (2000).

One obvious way out is to use different trees to impute each of the variables in Y, effectively turning a multivariate problem into q univariate ones. This is clearly undesirable, for it disregards relationships that may exists among components of Y; nonsensical imputations might be produced which fail to comply logical or arithmetic constraints that we know must hold.

To circumvent such problem, it is desirable to impute all variables in Y for each case i at once, taking the values of an observed "similar" case (again, multiple imputation is a possibility). This automatically guarantees consistency of the imputed values, and is a commonly accepted way to proceed (see Lejeune (1995), pág. 140 and Lebart and Lejeune (1995) in this connection). Section 3 describes the method we propose for multivariate imputation.

3 The intersection-seeking algorithm

3.1 Notation

To simultaneously impute $\mathbf{Y_i} = (Y_{i1}, \ldots, Y_{iq})$ we use the univariate trees $\mathcal{Y}_X^{(j)}$ constructed for each of the variables Y_j, $j = 1, \ldots, q$, as described next and formalized in Algorithm 2.

Let the nodes of each tree be numbered, and let $\mathcal{Y}_k^{(j)}$ be the k-th node of tree $\mathcal{Y}_X^{(j)}$. We use $\mathcal{Y}_k^{(j)}$ to denote the node, the subset of cases ending in, or going through, that node, and the corresponding region of $\mathcal{X}$. For instance, let $q = 2$ (i.e., there are two variables Y_1 e Y_2 in survey A) and let the trees $\mathcal{Y}_X^{(1)}$ and $\mathcal{Y}_X^{(2)}$ have the simple form depicted in Figure 2. Then, all cases in the training sample with $X_1 \leq a$ will end up in node $\mathcal{Y}_2^{(1)}$ when dropped down the tree constructed for variable Y_1; the corresponding region $\mathcal{Y}_2^{(1)}$ of $\mathcal{X}$ is shown in Figure 3.

For each q-tuple $(\alpha_1, \ldots, \alpha_q)$ such that α_j ($j \in \{1, \ldots, q\}$) is the label of a node in tree $\mathcal{Y}_X^{(j)}$, we define

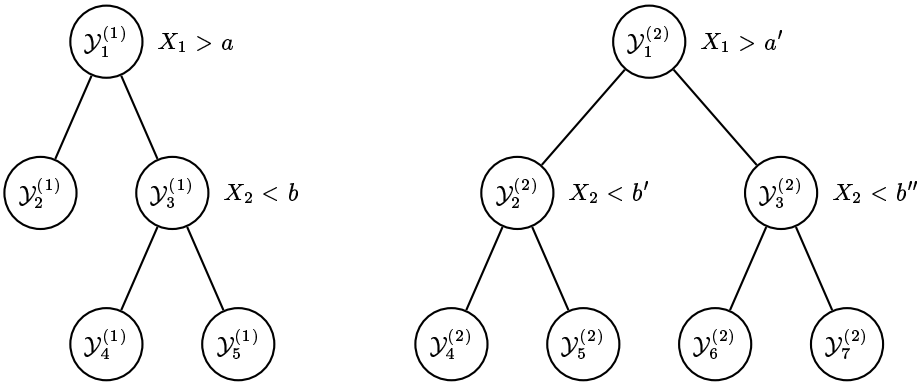
$$\mathcal{C}_{\alpha_1,\ldots,\alpha_q} \quad = \quad \mathcal{Y}_{\alpha_1}^{(1)} \cap \mathcal{Y}_{\alpha_2}^{(2)} \cap \ldots \cap \mathcal{Y}_{\alpha_q}^{(q)}. \tag{1}$$

Finally, let node $\mathcal{Y}_{(\uparrow\alpha_k)}^{(j)}$ be the "father" of node $\mathcal{Y}_{\alpha_k}^{(j)}$ in tree $\mathcal{Y}_X^{(j)}$. When there is no ambiguity about the tree referred to, we will simply refer to nodes $(\uparrow \alpha_k)$ and α_k.

3.2 Description

Consider now case i, $i \in \{N_A + 1, \ldots, N\}$, for which an imputation of $\mathbf{Y_i}$ is sought. Assume that when dropping that case through the trees built for each of the variables in

Figure 2: Trees $\mathcal{Y}_X^{(1)}$ and $\mathcal{Y}_X^{(2)}$. Next to each non terminal node is the condition whose fulfillment sends a case through the right son.



Y, it ends in the leaves $\mathcal{Y}_{i_1}^{(1)}, \ldots, \mathcal{Y}_{i_q}^{(q)}$ and hence belongs to $\mathcal{C}_{i_1,\ldots,i_q}$. The simple idea in our method is to impute $\mathbf{Y_i}$ as a function of the values $\mathbf{Y}$ from cases in the training sample (file A) which also belong to $\mathcal{C}_{i_1,\ldots,i_q}$. Those cases have values for each variable $Y_1, \ldots, Y_q$ which, as far as the relevant trees can ascertain, are indistinguishable from the ones of the case to impute.

As in the univariate Algorithm 1, a variety of options exist: to impute using one or several $\mathbf{Y}$ sampled randomly from $\mathcal{C}_{i_1,\ldots,i_q}$, using the mean, the median, or any other suitable function.

In the previous example, consider a case to impute i such that $a' < X_1 < a$ and $X_2 < b''$; it will end in leaves $\mathcal{Y}_2^{(1)}$ and $\mathcal{Y}_7^{(2)}$ when dropped down the trees $\mathcal{Y}_X^{(1)}$ and $\mathcal{Y}_X^{(2)}$. The intersection of those leaves,

$$\mathcal{C}_{2,7} = \mathcal{Y}_2^{(1)} \cap \mathcal{Y}_7^{(2)}, \tag{2}$$

is shown in Figure 4. We propose to impute $\mathbf{Y_i}$ using the values of $\mathbf{Y}$ observed for cases in the training sample that also fall in $\mathcal{C}_{2,7}$.

3.3 Details of the implementation

A problem may arise if case i to be imputed belongs to an intersection $\mathcal{C}_{i_1,\ldots,i_q}$ which is empty; no cases in the training sample belong to that particular intersection. When this happens, the intersection needs to be gradually enlarged to a non empty set: starting from the leaves $\mathcal{Y}_{i_1}^{(1)}, \ldots, \mathcal{Y}_{i_q}^{(q)}$ where i ended, our algorithm "climbs" the trees, replacing one node at a time by its "father". In doing so, we have at each step a choice of q trees that we may climb. The goal is to choose at each step in such a way that the quality of the imputation suffers least.

Let us see the heuristics implemented in our algorithm, which is one possible way of doing it. Continuous variables Y are assumed, but the idea can be generalized.

In the construction of trees, nodes are divided for as long this improves the fit in terms of *deviance* —for regression trees, usually the sum of squares is used; see

Figure 3: Partitions of the $\mathcal{X}$ space induced by trees $\mathcal{Y}_X^{(1)}$ and $\mathcal{Y}_X^{(2)}$.

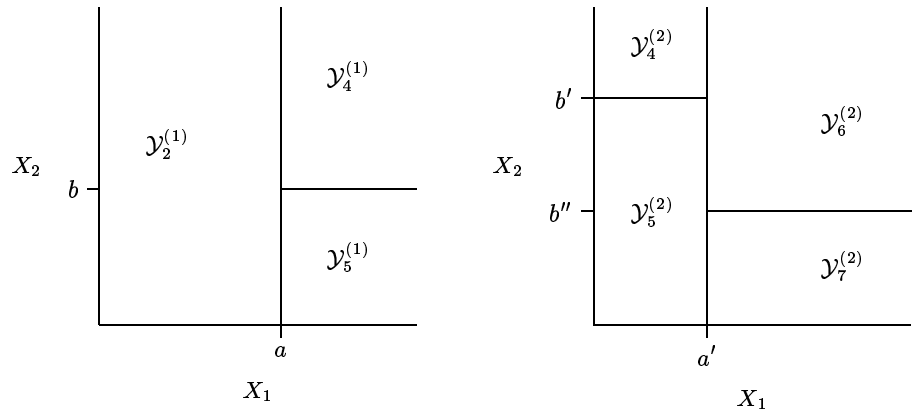
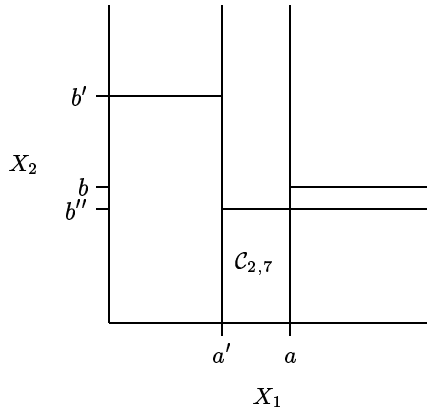


Figure 4: Overlay of partitions of $\mathcal{X}$ induced respectively by trees $\mathcal{Y}_X^{(1)}$ and $\mathcal{Y}_X^{(2)}$, and intersection $\mathcal{C}_{2,7}$.



Breiman et al. (1984), Cap. 3. Let $R(t)$ be the deviance at node t and $R(T)$ the total deviance of tree T, defined as

$$R(t) = \sum_{i \in t} (y_i - \overline{y}_t)^2 \tag{3}$$

$$R(T) = \sum_{t \in \tilde{T}} R(t), \tag{4}$$

where $\tilde{T}$ is the set of "leaves" or terminal nodes of tree T and $\overline{y}_t$ is the arithmetic mean of values of the response variable for the cases in node t.

For any of the trees $\mathcal{Y}_X^{(j)}$, $j = 1, \ldots, q$, the cost of climbing from node t_h to its father node t_p can be evaluated by

$$c^{(j)}(t_h) = \frac{\sum_{i=1}^{N_p} (y_{ij} - \overline{y}_{j,t_p})^2}{N_p} - \frac{\sum_{i=1}^{N_h} (y_{ij} - \overline{y}_{j,t_h})^2}{N_h} \tag{5}$$

$$= \hat{R}(t_p)/N_p - \hat{R}(t_h)/N_h \qquad \forall j = 1, .., q, \tag{6}$$

where $\hat{R}(t_h)$ and $\hat{R}(t_p)$ are resubstitution estimates of the deviance in node t_h (from which we consider climbing) and its father node t_p, N_p and N_h are the number of cases in nodes t_p and t_h respectively, and $\overline{y}_{j,t_p}$, $\overline{y}_{j,t_h}$ the means of variable Y_j for said nodes.

With the previous notation, we can specify Algorithm 2. A few comments are worth

Algorithm 2 – Multivariate imputation using trees.

1: *(optionally)* Compute suitable transformations of the variables Y.
2: Construct trees $\mathcal{Y}_X^{(1)}, \ldots, \mathcal{Y}_X^{(q)}$.
3: **for** $i \in$ {Cases to impute} **do**
4: Drop case i down the trees, and determine the intersection $\mathcal{C}_{\alpha_1,\ldots,\alpha_q}$ of the leaves $\mathcal{Y}_{\alpha_1}^{(1)}, \ldots, \mathcal{Y}_{\alpha_q}^{(q)}$ where it falls.
5: **while** $\mathcal{C}_{\alpha_1,\ldots,\alpha_q} = \emptyset$ **do**
6: Compute the costs $c^{(1)}(\alpha_1), \ldots, c^{(q)}(\alpha_q)$ of climbing from the current nodes.
7: Select k such that climbing from node α_k is of minimal cost.
8: $\alpha_k \leftarrow \quad (\uparrow \alpha_k)$; replace node α_k by its father.
9: **end while**
10: Impute i from $\mathcal{C}_{\alpha_1,\ldots,\alpha_q}$.
11: **end for**
12: *(if required)* Reconstruct the original variables from the imputed tranformations.

making. First, the criterion to choose which tree must be climbed is scale dependent. Therefore, we may want to scale the variables to have common variance, or variances which reflect their importance.

Second, Algorithm 2 can be applied both to the original variables or to any suitable transformations (for instance, principal components). The motivation behind using principal components is to reduce the number of trees to construct: this speeds up the process of finding an intersection. But both the first and last step in the algorithm are however optional.

4 Simulations

We tested the performance of the proposed method carrying out some simulations. We report below the results obtained in only a few situations, which give a flavour of the rest and bring out a number of interesting features.

4.1 Setup common to all simulations

We generated artificial samples from the general location model (see Olkin and Tate (1961), Schafer (1997)). We denote by $(\boldsymbol{W}, \boldsymbol{Z}) = (W_1, \ldots, W_r, Z_1, \ldots, Z_s)$ a vector with $(W_1, \ldots, W_r)$ categorical and $(Z_1, \ldots, Z_s)$ continuous variables. Variable W_ℓ can take d_ℓ levels, $\ell = 1, \ldots, r$. The categorical vector $\boldsymbol{W}$ can take $D = \prod_{\ell=1}^{r} d_\ell$ levels; its sample information can be summarized in a contingency table with that many cells. The distribution of $\boldsymbol{W}$ is fully specified by $\boldsymbol{\Pi} = (\pi_1, \ldots, \pi_D)$, where $\pi_d = \text{Prob}\{\boldsymbol{W} = \boldsymbol{w}_d\}$ and $\boldsymbol{w}_d$ denotes the values of $\boldsymbol{W}$ in cell d. Conditional on $\boldsymbol{W}$,

$$(\boldsymbol{Z}|\boldsymbol{W} = \boldsymbol{w}_d) \sim N(\boldsymbol{\mu_d}, \Sigma),$$

i.e. to each "state" $\boldsymbol{w}_d$ of $\boldsymbol{W}$ there is a conditional normal distribution for $\boldsymbol{Z}$ with a (possibly) different mean vector $\boldsymbol{\mu_d}$ and common covariance matrix Σ.

We considered situations with categorical and continuous predictors and responses always continuous. The p fully observed variables (the X columns in Figure 1) were made of p_{CAT} categorical variables from $\boldsymbol{W}$ and p_{CON} continuous variables from $\boldsymbol{Z}$. The q responses (the Y columns in Figure 1) were generated as functions of the $p = p_{\text{CAT}} + p_{\text{CON}}$ predictors, and then normal noise added. For instance, with $p_{\text{CAT}} = 3$ categorical, $p_{\text{CON}} = 2$ continuous and $q = 3$ responses, the "raw" responses were generated as

$$\begin{pmatrix} S_3 \\ S_4 \\ S_5 \end{pmatrix} \leftarrow \begin{pmatrix} 5 & 4 & 3 & 2 & 1 \\ 4 & 5 & 4 & 3 & 2 \\ 3 & 4 & 5 & 4 & 3 \end{pmatrix} \begin{pmatrix} W_1 \\ W_2 \\ W_3 \\ Z_1 \\ Z_2 \end{pmatrix}. \tag{7}$$

The raw responses where centered and scaled, multiplied by a factor SNR to adjust the signal-to-noise ratio —the ratio of the variance of the signal to the variance of the noise—, and noise added.

$$\begin{pmatrix} Z_3 \\ Z_4 \\ Z_5 \end{pmatrix} \leftarrow \text{SNR} \times \text{scaled} \begin{pmatrix} S_3 \\ S_4 \\ S_5 \end{pmatrix} + \begin{pmatrix} e_3 \\ e_4 \\ e_5 \end{pmatrix}$$

where:

$$\begin{pmatrix} e_3 \\ e_4 \\ e_5 \end{pmatrix} \sim N\left(\begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 & \gamma_q & \gamma_q^2 \\ \gamma_q & 1 & \gamma_q \\ \gamma_q^2 & \gamma_q & 1 \end{pmatrix} \right)$$

and

$$\begin{pmatrix} Z_1 \\ Z_2 \end{pmatrix} \sim N\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 & \gamma_p \\ \gamma_p & 1 \end{pmatrix} \right).$$

We considered the combinations in the number of predictors and responses and other parameters summarized in Table 1. At the chosen values of SNR $= 3, 5$ y 10, the

Table 1: Summary of simulation setup. Parameters in curly brackets {} represent alternatives. Each row of the table corresponds to 36 different simulation runs.

p_{CAT}	p_{CON}	q	Levels of $\boldsymbol{W}$	γ_p	γ_q	SNR	N
3	2	3	(3,2,3)	$\{0.01, 0.9\}$	$\{0.01, 0.9\}$	$\{3, 5, 10\}$	$\{200, 500, 1000\}$
5	0	5	(8,4,4,3,2)	$\{0.01, 0.9\}$	$\{0.01, 0.9\}$	$\{3, 5, 10\}$	$\{200, 500, 1000\}$
5	5	5	(8,4,4,3,2)	$\{0.01, 0.9\}$	$\{0.01, 0.9\}$	$\{3, 5, 10\}$	$\{200, 500, 1000\}$

correlations γ_p and γ_q showed no discernible influence, so we do not discuss them any further.

The functional relationship $f(\boldsymbol{W}, Z_1, \ldots, Z_{p_{\text{CON}}})$ linking predictors to responses was either linear with a coefficient matrix patterned as in (7) or quadratic.

When $f(\boldsymbol{W}, Z_1, \ldots, Z_{p_{\text{CON}}})$ is linear, the artificial data are generated according to the general location model: there is little doubt that a parametric imputation method which chooses the right model will work best. Our objective was to benchmark the intersection-seeking algorithm and see how much one looses in exchange for the flexibility and relative generality of a non parametric method.

We ported the library `mix` by J. Schafer (available at `http://www.stat.psu.edu/~jls/` writen for the S-Plus programming language) to R (described in Ihaka and Gentleman (1996)). This library includes functions for imputation based on the general location model. Functions were writen also in R for the intersection-seeking method, making extensive use of the library `rpart` (described in Therneau and Atkinson (1997) and available from CRAN, `http://cran.at.r-project.org`).

We wanted to see how well the missing information can be recovered by each method, rather than generate multiple imputations. From the total sample size N, the last 50 observations of the q responses were put aside and reconstructed once using each of the methods trained on the remaining $N-50$ observations. With the `mix` library, the imputations where values drawn from the distribution $f(\mathbf{X}_{\text{mis}}|\mathbf{X}_{\text{obs}}, \hat{\boldsymbol{\theta}}_{\boldsymbol{ML}})$, not taking into account the variability of $\hat{\boldsymbol{\theta}}_{\boldsymbol{ML}}$ (obtained with the EM algorithm). Hence, they are not "bayesianly proper" imputations (see Schafer (1997), p. 105).

When using the intersection-seeking method, we imputed once by drawing a single observation from the intersection $\mathcal{C}_{\alpha_1,\ldots,\alpha_q}$ chosen by Algorithm 2.

Each combination of parameters picking one from each column of Table 1 was used to generate $n = 500$ artificial samples.

4.2 Results

In the following we report on a subset of results which convey the essential of what we found.

The 50 observations deleted from each artificial sample were reconstructed using two methods: our intersection-seeking algorithm (Inter) and the EM plus data augmentation as implemented in the `mix` library referred to above. Table 2 lists the average square root of the mean square error (RMSE) of approximation for different combinations of signal-to-noise ratio (SNR) and sample size.

Four different cases are considered: the parameters and relationship between predictors and responses are given at the bottom of each column. We offer some comments in the following.

Case 1 is a small sized design with five predictors (three categorical, two normal) and three responses. The nature of the dependency among predictors and responses is linear. Thus, the general location model is adequate and given the small total number of parameters involved it is expected that the EM estimation plus imputation by data augmentation (in `mix`) will perform quite well. This is indeed the case. Notice that since we are imputing missing values by another like value (not by an estimated mean!) and the data are generated with variance equal to 1, the optimal RMSE is $\sqrt{2}$ (corresponds to imputing one missing value with another of exactly the same mean).

The value in the `mix` column for Case 1 nearly achieves this theoretical best performance, as could be expected. The small number of parameters are estimated quite precisely.

Table 2: Average RMSE of imputation. Each figure is the average on $n = 500$ replications. The parameters $p_{\text{CAT}}, p_{\text{CON}}$ and q and the type of dependency among predictors and responses is given below each column. $\sqrt{2(\text{SNR}^2 + 1)}$ is the RMSE achieved imputing with a case randomly chosen among those completely observed.

SNR	**N**	**Case 1**		**Case 2**		**Case 3**		**Case 4**	
		Inter	**mix**	**Inter**	**mix**	**Inter**	**mix**	**Inter**	**mix**
3	200	2.14	1.41	4.95	4.58	–	–	2.14	2.97
3	500	1.94	1.42	3.84	4.61	2.32	2.57	1.90	2.60
3	1000	1.92	1.43	3.40	4.56	2.32	2.13	1.92	2.15
$\sqrt{2(\text{SNR}^2+1)}$		4.47		4.47		4.47		4.47	
5	200	3.05	1.43	8.00	7.40	–	–	3.06	4.66
5	500	2.67	1.42	5.90	7.25	3.39	3.94	2.59	3.96
5	1000	2.59	1.42	5.25	7.40	3.38	3.04	2.61	3.02
$\sqrt{2(\text{SNR}^2+1)}$		7.21		7.21		7.21		7.21	
10	200	5.67	1.41	16.10	14.80	–	–	5.63	9.13
10	500	4.77	1.42	11.90	14.60	6.44	7.60	4.58	7.60
10	1000	4.57	1.42	10.40	14.60	6.35	5.57	4.59	5.56
$\sqrt{2(\text{SNR}^2+1)}$		14.21		14.21		14.21		14.21	
p_{CAT}		3		3		5		5	
With levels:		$(3,2,3)$		$(3,2,3)$		$(8,4,4,3,2)$		$(8,4,4,3,2)$	
p_{CON}		2		2		5		0	
q		3		3		5		5	
Dependency:		Linear		Quadratic		Linear		Linear	

The intersection-seeking algorithm does not perform nearly as well, although it is clearly better than random imputation.

The situation is reversed in Case 2. Here we have again the same (small) number of parameters, but the functional relationship linking responses and predictors is no longer linear, but quadratic. As expected, the general location model cannot cope with this and its RMSE of imputation is no better than random deck imputation.

The performance of the intersection-seeking algorithm also suffers, but is still better than random imputation, except for the smallest sample size: the trees are flexible enough to capture at least partially the relationship among predictors and responses.

We have found that even when responses and predictors are related as the general location model assumes, the intersection seeking algorithm may perform equally or better. The last two columns of Table 2 epitomize two such situations.

Case 3 considers the case with ten predictors (five categorical, five multivariate normal) and five multivariate normal responses. The smallest sample size ($N = 200$) has been dropped from the simulation as it was insufficient to use either method.

Even though the observations have been generated acording to the general location model, the non-parametric, intersection-seeking algorithm does nearly as well for the largest sample size ($N = 1000$) and looks even slightly better for $N = 500$. The reason for this seemingly couterintuitive result is the following: $p_{\text{CAT}} = 5$ categorical predictors with respectively 8, 4, 4, 3 and 2 levels give a total of $D = \prod d_\ell = 768$ cells. The general location model prescribes one mean vector $\boldsymbol{\mu_d}$ for each cell, totally unrelated to each other.

Now, clearly with $N = 500$ observations, a large portion of those mean vectors cannot be estimated. The `mix` library replaces the global mean vector $\boldsymbol{\mu}$ when there is need to impute a case with a combination of the p_{CAT} levels not seen in the training sample. No advantage is taken of the fact that, perhaps, a "similar" though not equal combination of levels is present in the training sample.

As compared to this, the intersection-seeking algorithm imputes from a pool of similar cases in an intersection of leaves. If the intersection is empty in the first instance, it will be enlarged gradually and a case be drawn from the first non empty intersection, rather than from the whole training sample; this accounts for its superiority when N is not large relative to $D = \prod d_\ell$.

This superiority is all the more noticeable when there are no continuous predictors. The general location model has a clear advantage at capturing linear relationships among continuous variables. Trees can only give a coarser, step-like approximation to those relations. When there are no continuous predictors and N is not large relative to D (Case 4 in Table 2), the intersection-seeking algorithm performs at its best, although the general location model recovers some ground as the sample size increases.

5 Summary and conclusions

A new method for imputation has been presented. It can cope with a large variety of problems, because of the generality of the tool used for approximation —classification or regression trees. It makes few assumptions, is computationally feasible, and appears to give good results: in simulated data, the method works well whenever the common variables X are good predictors for the Y's (see Figure 1) and the functional relationship among predictors and responses can be reasonably well approximated by a tree.

The method has been tested on simulated and real data sets of relatively large size (see Bárcena and Tusell (1999), Bárcena and Tusell (2000)) and can also be extended to cope with irregular patterns of missingness in the data (see Bárcena (2000)).

We see room for improvement, specially in the climbing strategy, at the expense of increased complexity and computational burden. Our work proceeds along this line. Further work is also required in comparing our method to other flexible, all-purpose methods of imputation, like those using neural networks.

References

Bishop, C. (1996). *Neural Networks for Pattern Recognition*. Oxford: Clarendon Press.

Bárcena, M. (2000). *Técnicas multivariantes para el enlace de encuestas*. Ph.D. thesis, Universidad del País Vasco.

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

Bárcena, M. and Tusell, F. (2000). Tree-based algorithms for missing-data imputation. In J. Bethlehem and P. van der Heijden, editors, *COMPSTAT'2000. Proceedings in Computational Statistics*, Heidelberg: Physica-Verlag.

Breiman, L., Friedman, J., Olshen, R., and Stone, C. (1984). *Classification and Regression Trees*. Wadsworth.

Buck, S. (1960). A method of estimation of missing values in multivariate data suitable for use with an electronic computer. *Journal of the Royal Statistical Society, Ser. B*, 22, 302–306.

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

Dempster, A., Laird, N., and Rubin, D. (1976). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Ser. B*, 39, 1–38.

Ihaka, R. and Gentleman, R. (1996). R: a language for data analysis and graphics. *J. of Comp. and Graphical Stats.*, 5, 299–314.

Lebart, L. and Lejeune, M. (1995). Assessment of Data Fusions and Injections. In *Encuentro Internacional AIMC sobre Investigación de Medios*, pp. 1–18, Madrid.

Lejeune, M. (1995). De l'usage des fusions de données dans les études de marché. In *Proceedings of the IASS Meeting, Beijing*.

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

Mesa, D., Tsai, P., and Chambers, R. (2000). Using tree based models for missing data imputation: and evaluation using UK Census data. Technical report, Department of Social Statistics, University of Southampton.

Nordbotten, S. (1996). Neural Network Imputation Applied to the Norwegian 1990 Population Census Data. *Journal of Official Statistics*, 12, 385–401.

Nordholt, E. (1998). Imputation: Methods, Simulation Experiments and Practical Examples. *International Statistical Review*, 66, 157–180.

Olkin, I. and Tate, R. (1961). Multivariate correlation models with mixed discrete and continuous variables. *Annals of Math. Stat.*, 32, 448–465.

Oudshoorn, K., van Buuren, S., and van Rijckevorsel, J. (1999). Flexible multiple imputation by chained equations of the AVO-95 Survey. Technical Report PG/VGZ/99.045, TNO Preventions and Health, Public Health, POB 2215, 2301 CE Leiden, Available from `http://www.multiple-imputation.com`.

Ripley, B. (1996). *Pattern Recognition and Neural Networks*. Cambridge University Press, 519.237.8 RIP.

Rubin, D. (1986). Statistical Matching Using File Concatenation With Adjusted Weights and Multiple Imputations. *Journal of Business and Economic Statistics*, 4, 87–94.

Rubin, D. (1987). *Multiple Imputation for Nonresponse in Surveys*. Wiley, 519.243 RUB.

Rubin, D. (1991). EM and beyond. *Psychometrika*, 56, 241–254.

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

Therneau, T. and Atkinson, E. (1997). An Introduction to Recursive Partitioning unsing the RPART Routines. Technical report, Mayo Foundation.

van Buuren, S. and Oudshoorn, C. (2000). Multivariate Imputation by chained equations. MICE V1.0 User's Manual. Technical report, TNO Prevention and Health.

van Buuren, S. and Oudshoorn, K. (1999). Flexible multiple imputation by MICE. Technical Report PG/VGZ/99.054, TNO Prevention and Health, Public Health, POB 2215, 2301 CE Leiden, Available from `http://www.multiple-imputation.com`.

Villagarcía, T. and Muñoz, A. (1997). Imputación de datos censurados mediante redes neuronales: una aplicación a la EPA. *Cuadernos Económicos de I.C.E.*, pp. 193–204.