Handling missing data in trees: surrogate splits or statistical imputation ?

Ad Feelders

Tilburg University
CentER for Economic Research
PO Box 90153
5000 LE Tilburg, The Netherlands
e-mail: A.J.Feelders@kub.nl

Abstract. In many applications of data mining a - sometimes considerable - part of the data values is missing. This may occur because the data values were simply never entered into the operational systems from which the mining table was constructed, or because for example simple domain checks indicate that entered values are incorrect. Despite the frequent occurrence of missing data, most data mining algorithms handle missing data in a rather ad-hoc way, or simply ignore the problem.
We investigate simulation-based data augmentation to handle missing data, which is based on filling-in (imputing) one or more plausible values for the missing data. One advantage of this approach is that the imputation phase is separated from the analysis phase, allowing for different data mining algorithms to be applied to the completed data sets.
We compare the use of imputation to surrogate splits, such as used in CART, to handle missing data in tree-based mining algorithms. Experiments show that imputation tends to outperform surrogate splits in terms of predictive accuracy of the resulting models. Averaging over $M > 1$ models resulting from M imputations yields even better results as it profits from variance reduction in much the same way as procedures such as bagging.

1 Introduction

It is generally recognized that data quality is a point of major concern in the construction of data warehouses, and subsequent analyses ranging from simple queries to data mining. The quality of knowledge extracted with data mining algorithms is evidently largely determined by the quality of the underlying data.

One important aspect of data quality is the proportion of missing data values. In many applications of data mining a - sometimes considerable - part of the data values is missing. This may occur because the data values were simply never entered into the operational systems from which the mining table was constructed, or because for example simple domain checks indicate that entered values are incorrect. Another common cause of missing data is the joining of not entirely matching data sets, which tends to give rise to monotone missing data patterns. Despite the frequent occurrence of missing data, many data mining algorithms handle missing data in a rather ad-hoc way, or simply ignore the problem.

Furthermore, the assumptions underlying the way missing data are handled are often not clear, which may lead to erroneous results if the implicit assumptions are violated [FCM98]. In this paper we focus on the well-known tree-based algorithm CART [BFOS84], that handles missing data by so called surrogate splits[1].

As an alternative we investigate more principled simulation-based approaches to handle missing data, based on filling-in (imputing) one or more plausible values for the missing data. One advantage of this approach is that the imputation phase is separated from the analysis phase, allowing for different data mining algorithms to be applied to the completed data sets.

2 Tree-based methods and missing data

Some well-known tree-based algorithms, such as CART [BFOS84] and C4.5 [Qui93], have ad-hoc procedures built-in to handle missing data. CART uses so-called *surrogate splits*, whereas C4.5 uses

[1] In fact we used the S program RPART that reimplements many of the ideas of CART, in particular the way it handles missing data.

fractional cases. In this section we shortly discuss the idea of surrogate splits, since we use the CART-like algorithm RPART in the experiments in section 4.

In tree-based algorithms the quality of a split s is defined as the reduction of impurity that it achieves, i.e.

$$\Delta i(s,t) = i(t) - p_R i(t_R) - p_L i(t_L), \tag{1}$$

where $i(t)$ denotes the impurity at any node t, p_R is the proportion of cases from t sent by s into t_R (the right child of t) and p_L is the proportion of cases from t sent into t_L (the left child of t). Well-known measures of impurity are the gini-index and the entropy measure.

We now have to consider how the quality of a split should be determined in the presence of missing values, and - once the best split is determined - which way to send a case with a missing value for that split. CART and RPART simply ignore missing values in determining the quality of a split, i.e. all quantities on the right-hand side of 1 are calculated from the non-missing values only.

In determining whether to send a case with a missing value for the best split left or right, the algorithm uses surrogate splits. It calculates to what extend alternative splits resemble the best split in terms of the number of cases that they send the same way. This resemblance is calculated on the cases with both the best split and alternative split observed.

Any observation with a missing value for the best split is then classified using the first (most resembling) surrogate split, or if that value is missing also, the second surrogate split, and so on. If an observation is missing all the surrogate splits then the default rule $\max(p_L, p_R)$ is used, i.e. the case is simply sent to the child with the largest relative frequency at that node. For more details, and subtle differences between CART and RPART see [BFOS84,TA97].

3 Multiple imputation

Multiple imputation [Sch97,Rub96] is a simulation-based approach where a number of complete data sets are created by filling in alternative values for the missing data. The completed data sets may subsequently be analyzed using standard complete-data methods, after which the results of the individual analyses are combined in the

appropriate way. The advantage, compared to using missing-data procedures tailored to a particular algorithm, is that one set of imputations can be used for many different analyses. The hard part of this exercise is to generate the imputations which may require computationally intensive data augmentation algorithms [Sch97,Tan96]. Multiple imputation was originally conceived to handle the problem of missing data in public-use databases where the database constructor and the ultimate user are distinct entities [Rub96]. In that situation it has clear advantages because most users have limited knowledge of models for missing data, and usually only have access to software for analysis of complete data sets. The situation is somewhat analogous for data warehouses in large organizations. A multitude of different analyses is performed on the data warehouse by different users, so it would be beneficial to solve the missing-data problem once, and allow the end-user to use his or her preferred complete-data analysis software.

4 Experiments

In our experiments we used software written in S-plus by J.L. Schafer[2] to generate the imputations. Since the examples we consider in this section contain both categorical and continuous variables we used the program MIX, which is based on the general location model. Basically the general location model is an extension of the well-known linear discriminant model to more than one categorical variable. For example if we have 2 categorical variables with 3 and 4 possible values respectively, then the categorical part is modeled by a multinomial distribution with 3×4 categories. Within each of these 12 categories the continuous variables are assumed to be normally distributed, where the means may differ between the categories but the covariance matrix is assumed to be equal. For a more detailed description of the general location model we refer the reader to chapter 9 of [Sch97]. The Bayesian nature of multiple imputation requires the specification of a prior distribution for the parameters of the imputation model. We used the default priors provided by the MIX program

[2] this software is available at http://stat.psu.edu/~jls/misoftwa.html

which are *non-informative*, i.e. correspond to a state of prior ignorance about the model parameters.

One of the critical parts of using multiple imputation is to assess the convergence of data augmentation. In our experiments we use the following simple rule of thumb suggested by Schafer [SO98]. Experience shows that data augmentation nearly always converges in fewer iterations than EM. Therefore we first computed the EM-estimates of the parameters, and recorded the number of iterations, say k, required. Then we perform a single run of the data augmentation algorithm of length $2Mk$, using the EM-estimates as starting values, where M is the number of imputations required. Just to be on the "safe side", we used the completed data sets from iterations $2k, 4k, \ldots, 2Mk$.

4.1 Waveform recognition data

To compare the performance of imputation with surrogate splits, we first consider an artificial data set used extensively in [BFOS84]. We summarize the properties of this classification problem below. There are three classes of observations, and each class is a convex combination of the three basic waveforms

$$h_1(t) = (0,1,2,3,4,5,6,5,4,3,2,1,0,0,0,0,0,0,0,0,0)$$
$$h_2(t) = (0,0,0,0,0,0,0,0,0,0,1,2,3,4,5,6,5,4,3,2,1,0)$$
$$h_3(t) = (0,0,0,0,0,1,2,3,4,5,6,5,4,3,2,1,0,0,0,0,0).$$

The measurement vector has 21 continuous components, i.e. $\mathbf{x} = (x_1, \ldots, x_{21})$. Observations from class 1 are constructed as follows:

$$x_m = uh_1(m) + (1-u)h_2(m) + \varepsilon_m, \quad m = 1, \ldots, 21$$

with $\varepsilon_m \sim N(0,1)$ and $u \sim U(0,1)$. Observation from class 2 and 3 are constructed analogously using (h_1, h_3) and (h_2, h_3) respectively.

We use the general location model for generating the imputations. The only categorical variable is the class label, and all covariates are continuous, so basically we are using the well- known linear discriminant model. Note that the assumptions of the linear discriminant model are not correct here, because the distribution of the covariates within each class is not multivariate normal and furthermore the covariance structure differs between the classes. Still, the model may be "good enough" to generate the imputations.

In the experiments, we generated 300 observations (100 from each class) to be used as a training set, with different percentages of missing data in the covariates. Then we built trees as follows

1. On the incomplete training set, using surrogate splits.
2. On one or more completed data sets using (multiple) imputation.

In both cases the trees were built using 10-fold cross-validation to determine the optimal value for the complexity parameter (the amount of pruning), using the program RPART[3].

The error rate of the trees was estimated on an independent test set containing 3000 *complete* observations (1000 from each class). To estimate the error rate at each percentage of missing data, the above procedure was repeated 10 times and the error rates were averaged over these 10 trials.

In a first experiment, each individual data item had a fixed probability of being missing. Table 1 summarizes the comparision of surrogate splits and single imputation at different fractions of missing data. Single imputations are drawn from the predictive distribution of the missing data given the observed data and the EM-estimates for the model parameters. Looking at the difference between the error rates one can see that imputation gains an advantage when the level of missing data becomes higher. However, at a moderate level of missing data (say 10% or less) it doesn't seem worth the extra effort of generating imputations. This same trend is also clear from rows four (p^+_{imp}) and five (p^-_{imp}) of the table. p^+_{imp} (p^-_{imp}) indicates the number of times of the ten trials, that the error rate of imputation was higher (lower) and the difference was significant at the 5% level. So, for example, at 30% missing data the difference was significant at the 5% level four out of ten times, and in all four cases the error rate of imputation was lower.

In a second experiment we used multiple imputation with $M = 5$, and averaged the predictions of the 5 resulting trees. The results are given in table 2. The performance of multiple imputation is clearly better than both single imputation and surrogate splits. The average tree obtained by multiple imputation even has a lower error rate than the single tree on complete data (independent experiments showed

[3] RPART is written by T. Therneau and E. Atkinson in the S language. The S-plus version for Windows is available from http://www.stats.ox.ac.uk/pub/Swin.

% Missing	10	20	30	40	45
$\hat{e}_{sur}$	29.8%	30.9%	32.2%	32.4%	34.3%
$\hat{e}_{imp}$	29.8%	29.2%	30.6%	30.0%	30.4%
$\hat{e}_{sur} - \hat{e}_{imp}$	0%	1.7%	1.6%	2.4%	3.9%
p^{+}_{imp}	1	0	0	0	0
p^{-}_{imp}	1	4	4	6	7

Table 1. Estimated error rate of surrogate splits and single imputation at different fractions of missing data (estimates are averages of 10 trials)

that a single tree has an error rate of about 29%). Apparently this gain comes from the variance reduction resulting from averaging a number of trees, like is done in bagging [Bre96]. For comparison, we estimated the error rate of a bagged tree based on 5 *complete* bootstrap samples, which yielded an error rate of about 24%. As was to be expected this is lower than the error rate achieved with multiple imputation on *incomplete* data.

% Missing	10	20	30	40	45
$\hat{e}_{sur}$	28.9%	30.1%	30.0%	33.3%	35.6%
$\hat{e}_{imp}$	26.0%	26.1%	25.5%	25.7%*	26.0%*
$\hat{e}_{sur} - \hat{e}_{imp}$	2.9%	4.0%	4.5%	7.6%	9.6%
p^{+}_{imp}	0	0	0	0	0
p^{-}_{imp}	9	8	9	10	10

Table 2. Estimated error rate of surrogate splits and multiple imputation at different fractions of missing data. *: here we ran into problems with data augmentation and used EM-estimates only to generate the imputations

Finally, in a third experiment we considered a rather extreme case of a monotone missing data pattern. The only missing data pattern that occurs is that some observations have $x_3, \ldots, x_{21}$ missing. Such a pattern would typically occur when missing data is due to the coupling of incompletely matching tables. As noted by Breiman ([BFOS84], p. 146) one would expect the results of surrogate splits to be worse in this case because for each observation with missing values, the only surrogates available are x_1 and x_2.

The results are summarized in table 3. Up to 30% of missing data, the results are comparable to those of the non-monotone pattern of

% Missing	10	20	30	40	45
$\hat{e}_{sur}$	30.3%	30.2%	32.5%	34.8%	38.5%
$\hat{e}_{imp}$	29.4%	29.6%	31.3%	30.5%	30.7%
$\hat{e}_{sur} - \hat{e}_{imp}$	0.9%	0.6%	1.2%	4.3%	7.8%
p^+_{imp}	1	2	1	0	0
p^-_{imp}	4	4	4	8	10

Table 3. Estimated error rate of surrogate splits and single imputation at different fractions of missing data (monotone pattern)

table 1. Only at an extremely high level of montone missing data do the surrogate splits "break down" on this data set. The imputation trees appear to be quite robust with a mean error rate of 30.7% at 45% of missing data. An additional advantage of monotone patterns for imputation is that convergence of EM and data augmentation is relatively fast.

4.2 Pima indians database

In this section we perform a comparison of surrogate splits and imputation on a real life data set that has been used quite extensively in the machine learning literature. It is known as the *Pima Indians Diabetes Database*, and is available at the UCI machine learning repository [BKM99].

The class label indicates whether the patient shows signs of diabetes according to WHO criteria. Although the description of the dataset says there are no missing values, there are quite a number of observations with "zero" values that most likely indicate a missing value. In table 4 we summarize the content of the dataset, where we have replaced zeroes by missing values for $x_3, \ldots, x_7$. The dataset contains a total of 768 observations, of which 500 of class 0 and 268 of class 1.

In our experiment the test set consists of the 392 complete observations, and the training set consists of the remaining 376 observations with one or more values missing. Of these 376 records, 374 have a missing value for x_6 (serum insulin), so we removed this variable. Furthermore, we changed x_1 (number of times pregnant) into a binary variable indicating whether or not the person had ever been pregnant (the entire dataset consists of females at least 21 years old,

Variable	Description	Missing values
y	Class label (0 or 1)	0
x_1	Number of times pregnant	0
x_2	Age (in years)	0
x_3	Plasma glucose concentration	5
x_4	Diastolic blood pressure	35
x_5	Triceps skin fold thickness	227
x_6	2-hour serum insulin	374
x_7	Body mass index	11
x_8	Diabetes pedigree function	0

Table 4. Overview of missing values in pima indians database

so this variable is always applicable). This leaves us with a dataset containing two binary variables (y and x_1) and six numeric variables ($x_2, \ldots, x_5$, x_7 and x_8), with $278/2632 \approx 10\%$ missing values in the covariates. Although x_2 and x_8 are clearly skewed to the right, we did not transform them to make them appear more normal, in order to get an impression of the robustness of imputation under the general location model.

The first experiment compares the use of surrogate splits to imputation of a single value based on the EM-estimates. Of course the tree obtained after single imputation depends on the values imputed. Therefore we performed ten independent draws, to get an estimate of the average performance of single imputation. The results are summarized in table 5.

Draw	1	2	3	4	5	6	7	8	9	10
$\hat{e}_{imp}$	22.7%	30.6%	25.3%	26.0%	30.0%	24.5%	26.8%	24.7%	27.8%	29.3%
p-value	.0002	1	.0075	.0114	.7493	.0097	.0237	.0038	.2074	.6908

Table 5. Estimated error rates of ten single imputation-trees and the corresponding p-values of $H_0 : e_{imp} = e_{sur}$, with $\hat{e}_{sur} = 30.6\%$

For each single imputation-tree, we compared the performance on the test set with that of the tree built using surrogate splits, which had an error rate of $120/392 \approx 30.6\%$.

Tests of $H_0 : e_{sur} = e_{imp}$ against a two-sided alternative, using an exact binomial test, yield the p-values listed in the second row of

table 5. On average the single imputation-tree has an error rate of 26.8% which compares favourably to the error rate of 30.6% of the tree based on the use of surrogate splits.

In a second experiment we used multiple imputation ($M = 5$) and averaged the predictions of the 5 trees so obtained. Table 6 summarizes the results of 10 independent trials. The average error rate of the multiple imputation-trees over these 10 trials is approximately 25.2%. This compares favourably to both the single tree based on surrogate splits, and the tree based on single imputation.

Trial	1	2	3	4	5	6	7	8	9	10
$\hat{e}_{imp}$	27.3%	24.5%	25.8%	26.8%	23.7%	24.2%	24.0%	25.5%	24.7%	25.5%
p-value	.1048	.0015	.0295	.0357	.0003	.0026	.0022	.0105	.0027	.0119

Table 6. Estimated error rates of 10 multiple imputation-trees ($M = 5$), and the corresponding p-values of $H_0 : e_{imp} = e_{sur}$, with $\hat{e}_{sur} = 30.6\%$

5 Discussion and Conclusions

The use of statistical imputation to handle missing data in data mining has a number of attractive properties. First of all, the imputation phase and analysis phase are separated. Once the imputations have been generated the completed data sets may be analysed with any appropriate data mining algorithm. The imputation model does not have to be the "true" model (otherwise why not stick to that model for the complete analysis?) but should merely be good enough for generating the imputations. We have not performed systematic robustness studies, but in both data sets analysed the assumptions of the general location model were voilated to some extent. Nevertheless, the results obtained with imputation were nearly always better than those with surrogate splits. An advantage not explored in this study is that imputation is able to handle missing data in the dependent variable as well as in the covariates, unlike the procedures used by tree-based algorithms such as CART and C4.5.

Despite these theoretical advantages, one should still consider whether they outweigh the additional effort of specifying an appropriate imputation model and generating the imputations. From the

experiments we performed some tentative conclusions may be drawn. For the waveform data, single imputation tends to outperform surrogate splits as the amount of missing data increases. At moderate amounts of missing data (say 10% or less) one can avoid generating imputations and just use surrogate splits. For the pima indians data, with about 10% missing data in the training set, single imputation already shows a somewhat better predictive performance.

Multiple imputation shows a consistenly superior performance, as it profits from the variance reduction achieved by averaging the resulting trees. For high variance models such as trees and neural networks multiple imputation may therefore yield a substantial performance improvement.

References

[BFOS84] Leo Breiman, Jerome H. Friedman, Richard A. Olshen, and Charles T. Stone. *Classification and Regression Trees*. Wadsworth, Belmont, California, 1984.

[BKM99] C. Blake, E. Keogh, and C.J. Merz. UCI repository of machine learning databases. University of California, Irvine, Dept. of Information and Computer Sciences, 1999. http://www.ics.uci.edu/~mlearn/MLRepository.html.

[Bre96] L. Breiman. Bagging predictors. *Machine Learning*, 26(2):123–140, 1996.

[FCM98] A.J. Feelders, S. Chang, and G.J. McLachlan. Mining in the presence of selectivity bias and its application to reject inference. In R. Agrawal, P. Stolorz, and G. Piatetsky-Shapiro, editors, *Proceedings of the fourth international conference on knowledge discovery and data mining*, pages 199–203, Menlo Park, California, 1998. AAAI Press.

[Qui93] J. Ross Quinlan. *C4.5 Programs for Machine Learning*. Morgan Kaufmann, San Mateo, California, 1993.

[Rub96] D.B. Rubin. Multiple imputation after 18+ years. *Journal of the American Statistical Association*, 91:473–489, 1996.

[Sch97] J.L. Schafer. *Analysis of Incomplete Multivariate Data*. Chapman & Hall, London, 1997.

[SO98] J.L. Schafer and M.K. Olsen. Multiple imputation for multivariate missing-data problems: a data analyst's perspective. *Multivariate Behavioral Research*, 33(4):545–571, 1998.

[TA97] T.M. Therneau and E.J. Atkinson. An introduction to recursive partitioning using the RPART routines. Mayo Foundation, 1997.

[Tan96] M.A. Tanner. *Tools for Statistical Inference (third edition)*. Springer, New York, 1996.