

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. 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

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 they were simply never entered into the operational systems, 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, many data mining algorithms handle missing data in a rather ad-hoc way, or simply ignore the problem.

In this paper we focus on the well-known tree-based algorithm CART [3], that handles missing data by so called surrogate splits¹. As an alternative we

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.

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 Multiple Imputation

Multiple imputation [5,4] 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 algorithms such as data augmentation and Gibbs sampling [5,7].

In our experiments we used software for data augmentation written in S-plus by J.L. Schafer² to generate the imputations. Since the examples we consider in this section contain both categorical and continuous variables, imputations are based on the general location model (see [5], chapter 9). The Bayesian nature of multiple imputation requires the specification of a prior distribution for the parameters of the imputation model. We used a *non-informative* prior, i.e. a prior corresponding 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 used a rule of thumb suggested by Schafer [6]. 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 $2k, 4k, \dots, 2Mk$.

3 Waveform Recognition Data

To compare the performance of imputation with surrogate splits, we first consider the *waveform recognition data* used extensively in [3]. The only categorical variable is the class label (with 3 possible values), and all 21 covariates are continuous, so imputation is based on 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

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

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 RPART³.

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 comparison 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.

% 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)

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

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>.

performance of multiple imputation is clearly better than both single imputation and surrogate splits. Presumably, this gain comes from the variance reduction resulting from averaging a number of trees, like is done in bagging [2].

% 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

4 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 [1].

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 3 we summarize the content of the dataset, where we have replaced zeroes by missing values for x_3, \dots, x_7 . The dataset contains a total of 768 observations, of which 500 of class 0 and 268 of class 1.

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 3. Overview of missing values in pima indians database

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, 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, \dots, 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 4.

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 4. 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 4. 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 5 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.

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

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

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 violated to some extent. Nevertheless, the results obtained with imputation were nearly always better than those with surrogate splits.

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 consistently 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

1. 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>.
2. L. Breiman. Bagging predictors. *Machine Learning*, 26(2):123–140, 1996.
3. L. Breiman, J.H. Friedman, R.A. Olshen, and C.T. Stone. *Classification and Regression Trees*. Wadsworth, Belmont, California, 1984.
4. D.B. Rubin. Multiple imputation after 18+ years. *Journal of the American Statistical Association*, 91:473–489, 1996.
5. J.L. Schafer. *Analysis of Incomplete Multivariate Data*. Chapman & Hall, London, 1997.
6. 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.
7. M.A. Tanner. *Tools for Statistical Inference (third edition)*. Springer, New York, 1996.