A Study of K-Nearest Neighbour as a Model-Based Method to Treat Missing Data

Gustavo E.A.P.A. Batista and Maria Carolina Monard

University of São Paulo - USP
Institute of Mathematics Sciences and Computer - ICMC
Department of Computer Science and Statistics - SCE
Laboratory of Computational Intelligence - LABIC
P. O. Box 668, 13560-970 - São Carlos, SP, Brazil
{gbatista, mcmonard}@icmc.sc.usp.br

Abstract. The increasing interest in Knowledge Discovery from Databases (KDD) has made data quality a central topic in Machine Learning and other research areas of KDD, since many researches have reported that the quality of the data extracted from databases is not good. The data quality problems found in real databases are usually more complex than those found in data from repositories. In this way, some data pre-processing techniques broadly used by the Machine Learning community should be revised. An example is the missing data treatment done by some learning systems. Frequently, the unknown values of an attribute are substituted by the mean value (or the most frequent value) of the attribute. This sort of technique is only suitable for datasets with few and randomly distributed missing values. If these pre-conditions are not satisfied, invalid knowledge can be introduced into the data. Data with high level of missing data or with non-randomly distributed missing data should be treated by more robust methods, for instance model-based methods. Model-based methods consist in creating a predictive model to predict the values of the missing data. In this work we analyse the performance of the k-nearest neighbour algorithm as a model-based method to treat missing data. The performance of the k-nearest neighbour is compared to the performance of the internal algorithm used by the learning system C5.0 to treat missing data.

1 Introduction

For years, Machine Learning (ML) researchers have applied their algorithms to "ready-to-learn" datasets provided by data repositories. These datasets are already pre-processed, *i.e.*, most of the problems that can be found in "real world[1]" data have already been identified and corrected. The increasing interest in Knowledge Discovery from Databases (KDD) has made data quality a central topic in Machine Learning and other research areas of KDD, since many

[1] In this work, the terms "real data" and "real world data" refer to data collected from data storage systems. Data provided by data repositories such as UCI [6] are called "natural" [8, 4] to distinguish them from artificial datasets.

researches have reported that the quality of the data extracted from databases is not good. The data quality problems found in real databases are usually more complex than those found in data from repositories. In this way, some data preprocessing techniques broadly used by the Machine Learning community should be revised. An example is the missing data treatment done by some learning systems. Frequently, the missing values of an attribute are substituted by the mean value (or the most frequent value) of the attribute. This sort of technique is only suitable for datasets with few and randomly distributed missing values. If these pre-conditions are not satisfied, invalid knowledge can be introduced into the data.

Thus, the characteristics of missing data should be analysed in order to choose a proper treatment. Data with high level of missing values, or with non-randomly distributed missing values, should be treated by robust methods, for instance model-based methods. Model-based methods consist in creating a predictive model to predict values of missing data. In order to build such a model, the attribute with missing values is used as class (output attribute) and the other attributes are used as input.

This work is part of a research project about missing data treatment in Knowledge Discovery from Databases. One of the main objectives of this research is to identify in which situations missing data can be dangerous if not treated properly. In this way, Section 2 presents two characteristics of missing data that should be analysed in order to decide which method should be used to treat the problem. These characteristics are pattern and amount of missing data. Other objective of this research is to identify the most important methods for missing data treatment in the literature, aiming to identify when each method is more effective. Section 3 describes one of the most popular and robust class of methods to treat missing values, so called model-based methods. We are also developing a number of experimental analysis in order to associate missing data treatment methods with characteristics of the missing data. Section 4 presents some experiments to evaluate the efficiency of the k-nearest neighbour as a model-based method to treat missing data. These results are compared with the performance of the internal algorithm used for the same purpose by C5.0 [7]. C5.0 uses a probabilistic approach to handle missing values in the training and test data. In this work, we are mainly interested in analysing the behaviour of these two methods when they are used to treat a large amount of missing data, since many researchers of the KDD community [5] have reported to find more than 50% of missing data when extracting data from databases. Finally, Section 5 presents the conclusions of this work.

2 The Importance of Pattern and Amount of Missing Data

In a general way, the pattern of missing data is more important than the amount of missing data [9]. Missing data scattered randomly through a dataset can be considered a less serious problem than when those values are not randomly dis-

tributed. On the other hand, non-randomly missing values are a serious problem no matter how few they are, since these values may affect the generalizability of results. In this case, some methods for estimating missing values are needed. When only few examples have missing values in a large dataset and those values show a random pattern, then the missing values will likely influence very little the results. In this case, almost any procedure to handle missing data will provide similar results. In fact, simply removing the cases with missing data is one of the most simple and fast ways to solve the problem. However, if a dataset has a large amount of cases with missing data, the removal of those cases is likely to degrade the performance of the induced classifier. Unfortunately, there are no firm guidelines to tell us how much missing data can be tolerated for a sample of a given size [9].

3 Model-based Methods for Handling Missing Data

Model-based methods are a sophisticated procedure for handling missing data. These methods consist of creating a predictive model to estimate values that will substitute the missing data. The attribute with missing data is used as class-attribute, and the remaining attributes are used as input for the predictive model. An important argument in favour of this approach is that, frequently, attributes have relationships (correlations) among themselves. In this way, those correlations could be used to create a predictive model for classification or regression (depending on the attribute type with missing data, being, respectively, nominal or continuous). Some of these relationships among the attributes may be maintained if they were captured by the predictive model. An important drawback of this approach is that the model estimated values are usually more well-behaved than the true values would be, *i.e.*, since the missing values are predicted from a set of attributes, the predicted values are likely to be more consistent with this set of attributes than the true values (not known) would be. A second drawback is the requirement for correlation among the attributes. If there are no relationships among one or more attributes in the dataset and the attribute with missing data, then the data estimated by the model will not be precise.

In this work we propose the use of k-nearest neighbour algorithm to estimate and substitute missing data. The main benefits of this approach are:

- k-nearest neighbour can predict both discrete attributes (the most frequent value among the k nearest neighbours) and continuous attributes (the mean among the k nearest neighbours);
- There is no necessity for creating a predictive model for each attribute with missing data. Actually, the k-nearest neighbour does not create models (like a decision tree or a set of rules), once the training examples are used as a model. Thus, the k-nearest neighbour can be easily adapted to work with any attribute as class, by just modifying which attributes will be considered in the distance metric. Also, this approach can easily treat examples with multiple missing values.

The main drawback of this approach is:

– Whenever the k-nearest neighbour looks for the most similar examples, the algorithm searches through all the training examples. This limitation can be very critical for KDD, since this research area has, as one of its main objectives, the analysis of large databases. Many works that aim to solve this limitation can be found in the literature. One of the most well-known methods is the creation of a reduced training set for the k-nearest neighbour composed only by prototypical examples [10].

4 Experimental Analysis

The main objective of the experiments conducted in this work is to evaluate the efficiency of the k-nearest neighbour algorithm as a model-based method to treat missing data, and compare its performance with the performance obtained by the internal algorithm used by C5.0 to learn with missing data. C5.0 was chosen due to the fact that it is considered one of the best algorithms to induce propositional concepts. It uses a probabilistic approach to handle missing values. In our experiments, missing data was artificially implanted, in different rates and attributes, into the dataset. The performance of both missing data treatments are compared using cross-validation estimated error rates. In particular, we are interested in analysing the behaviour of these two treatments when the amount of missing data is high since, as stated before, many researchers have reported to find databases where more than 50% of the data was missing.

The experiments were done with the "Breast Cancer" dataset [6]. Table 1 shows some statistics of this dataset.

Number of attributes	9
Number of continuous attributes	9
Number of examples	699
Number of examples with missing data	16
Number of classes	2
Number of examples belonging to "benign" class	458 (65.52%)
Number of examples belonging to "malignant" class	241 (34.48%)

Table 1. Some statistics about the "Breast Cancer" dataset.

Initially, the original dataset is partitioned into 10 pairs of training and test sets through the application of 10-fold cross validation. The performance measures are done with the aid of the AMPSAM environment [2]. The steps shown in Figure 1 are done for each pair of training and test sets. Missing data is inserted into the training set. Two copies of this training set are used, one is given to the C5.0 without any missing data treatment. The other one is treated by applying the k-nearest neighbour to estimate and substitute missing data. After

the treatment of missing data, the training set is given to C5.0 and a classifier is induced. Both classifiers, *i.e.* the one induced with untreated data and the other one induced with treated data, are used to classify the test set. At the end of 10 iterations, we can estimate the true error rate by calculating the mean of the error rates of each iteration. Finally, the performance of C5.0 allied to the missing data treatment method can be analysed and compared to the performance of the method used by C5.0 to learn when missing data is present.

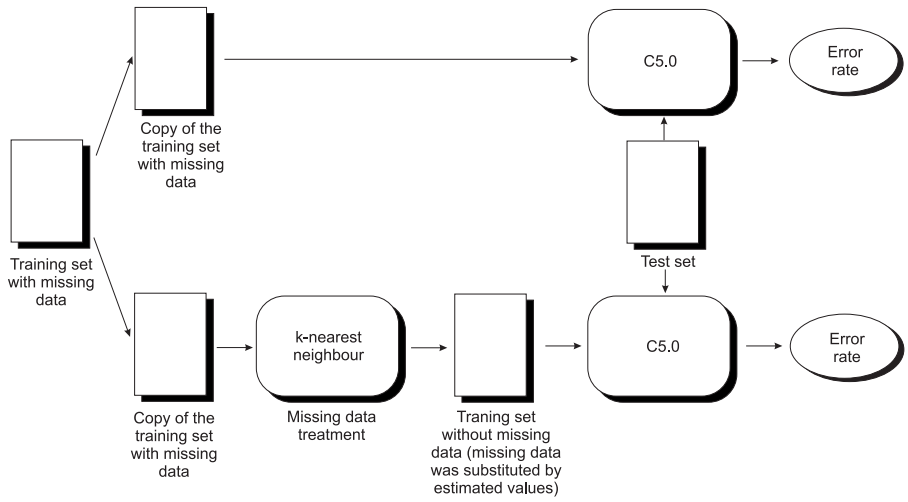

Fig. 1. Methodology used in the experiments.

Originally, the Breast Cancer dataset has 16 examples with missing data. These examples were removed from the dataset before the beginning of the experiments. In this way, the dataset was reduced to 683 examples without missing data. The main reason for the removal is that we want to have total control over the missing data in the dataset. For instance, we would like that the test sets do not have any missing data. If some test set has missing data, then the inducer's ability to classify missing data properly may influence on the result. This influence is undesirable since the objective of this work is to analyse the viability of the model-based missing data treatment, and the inducer learning ability when missing data is present.

The next step is to insert missing data in the training sets. In order to do that, some attributes have to be chosen, and some of their values modified to unknown. Which attributes will be chosen and how many of their values will be modified to unknown is an important decision. It is easy to see that the most representative attributes of the dataset are a sensible choice for the attributes that should have their values modified to unknown. Otherwise, the analysis may be compromised by treating non-representative attributes that will not be incorporated into the classifier by the learning system. Since to find the most representative attributes

of a dataset is not a trivial task, we decided to find out which attributes in the dataset are important for C5.0. In order to do so, C5.0 was run with all 683 examples and the 3 attributes nearest to the tree root were selected, they are: "Uniformity of Cell Size"; "Uniformity of Cell Shape" and "Bare Nuclei". There is no assurance that these attributes will be incorporated by C5.0 into the classifiers induced in the experiments. The existence of an attribute with similar information (high correlation) with one of the selected attributes, can make C5.0 choose not to use the selected attribute.

With relation to the amount of missing data to be inserted into the training sets, we want to analyse the behaviour of the methods with different amounts of missing data. In this way, missing data was randomly inserted with the following percentages: 10%, 20%, 30%, 40% and 50% for each attribute. For instance, when 20% of missing data is inserted in the attributes "Uniformity of Cell Size" and "Uniformity of Cell Shape", it means that 20% of the values of the attribute "Uniformity of Cell Size" were randomly selected and changed to the unknown value. After that, 20% of the values of the attribute "Uniformity of Cell Shape" were also randomly selected (independently of the previously selected values of the attribute "Uniformity of Cell Size") and changed to unknown.

The experiments were done with missing data inserted into all three selected attributes "Uniformity of Cell Size", "Uniformity of Cell Shape" e "Bare Nuclei" (Figure 2-a), with missing data inserted into the attributes "Uniformity of Cell Size" and "Uniformity of Cell Shape" (Figure 2-b), and with missing data inserted only into the attribute "Uniformity of Cell Size" (Figure 2-c). This last attribute was chosen by C5.0 as root of the tree when inducing over all examples). In our experiments, we initially tested different values for the parameter k (number of nearest neighbours) of the k-nearest neighbour algorithm. As the results were similar for all tested k values, we decided to continue the experiments using the parameter $k = 3$.

Considering the results shown in Figure 2-a, the performance of both methods is very similar, even when there exist a large amount of missing data. In a general way, the model-based method was slightly superior to the internal method used by C5.0, with exception of the error rate obtained with a training set with 30% of missing data. The result obtained with 30% of missing data seems not to agree with the other results in the graph.

Also, Figure 2-b shows that the model-based method performance is slightly superior to the results obtained without missing data treatment.

Finally, Figure 2-c presents the results obtained with missing data inserted only into the attribute "Uniformity of Cell Size". Interestingly, whenever missing data was present in the training set, C5.0 was able to obtain lower error rates than when no missing data was present. Also, the error rate for all classifiers created from a training set with missing data was the same (4.18%). After a more detailed analysis of the decision trees induced by C5.0, we observed that C5.0 was not using the attribute "Uniformity of Cell Size" in all induced classifiers whenever this attribute had missing data. On the other hand, in all occasions that the missing data was treated by our method, the attribute was incorporated

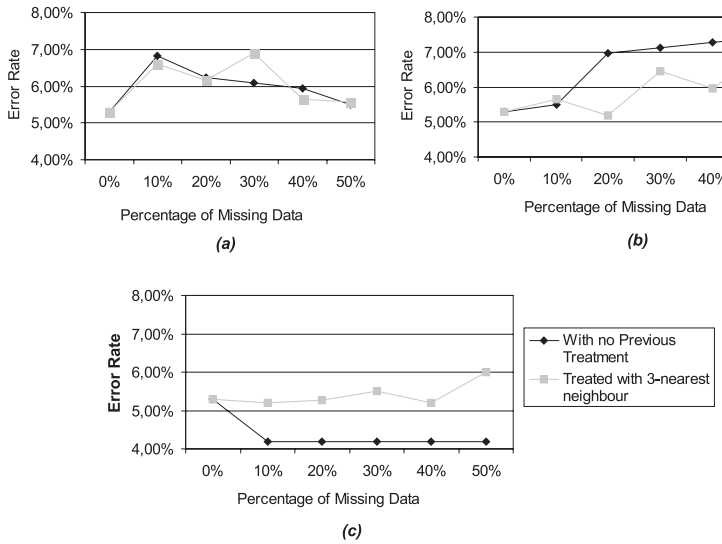

Fig. 2. Results shown graphically. Missing data inserted into the attributes "Uniformity of Cell Size", "Uniformity of Cell Shape" and "Bare Nuclei" (a); "Uniformity of Cell Size" and "Uniformity of Cell Shape" (b); and "Uniformity of Cell Size" (c).

to the induced classifier. C5.0 was able to create a classifier apparently more precise by ignoring the attribute "Uniformity of Cell Size", which was chosen as the decision tree root in our preliminary studies. As we previously observed, the method used to choose the most representative attributes to insert missing data into their values is not foolproof.

Table 2-(a, b and c) shows the numerical results of the graphs presented in Figure 2-(a, b and c), respectively, as well as the standard deviations. Analysing the results in Table 2-a, it can be observed that the error rates for both methods are few percentage points distant from the error rate obtained by the classifier induced without missing data (0%). This difference is small even when there exist a large amount of missing data (for instance, 40% and 50%). A statistical test can be used in order to verify (with 95% of confidence) if there is a significant difference among the error rate obtained by the classifier induced from a training set with 0% of missing data, and the error rates of the classifiers generated from a training set with missing data. In this work, we decided to use the hypothesis test described in [1]. If this hypothesis test gives a result, in modulus, equal or greater than 2 then there is a statistically significant difference between two error rates with 95% of confidence.

In our experiments, the error rate obtained by C5.0 and a training set free of missing data is compared with the error rates obtained by the two methods of missing data treatment (the C5.0 internal method for dealing with missing data and the model-based method using k-nearest neighbour) for the training

	%?	No Previous Missing Data Treatment	Hypothesis Test (No Previous Missing Data Treatment)	Missing Data Treated by 3-Nearest Neighbour	Hypothesis Test (Missing Data Treated)
	0%	5.30 ± 0.60	-	5.30 ± 0.60	-
	10%	6.82 ± 0.51	-2.73	6.60 ± 0.79	-1.85
(a)	20%	6.24 ± 0.63	-1.53	6.16 ± 0.94	-1.09
	30%	6.09 ± 0.58	-1.34	6.90 ± 0.73	-2.39
	40%	5.94 ± 0.65	-1.02	5.65 ± 0.84	-0.48
	50%	5.50 ± 0.69	-0.31	5.57 ± 0.59	-0.45
	0%	5.30 ± 0.60	-	5.30 ± 0.60	-
	10%	5.50 ± 0.81	-0.28	5.65 ± 0.56	-0.60
(b)	20%	6.98 ± 0.95	-2.11	5.20 ± 0.77	0.14
	30%	7.12 ± 0.98	-2.24	6.45 ± 0.81	-1.61
	40%	7.27 ± 0.92	-2.54	5.79 ± 0.59	-0.82
	50%	7.41 ± 0.92	-2.72	6.97 ± 0.69	-2.58
	0%	5.30 ± 0.60	-	5.30 ± 0.60	-
	10%	4.18 ± 0.56	1.93	5.20 ± 1.01	0.12
(c)	20%	4.18 ± 0.56	1.93	5.28 ± 0.68	0.03
	30%	4.18 ± 0.56	1.93	5.50 ± 0.78	-0.29
	40%	4.18 ± 0.56	1.93	5.20 ± 0.45	0.19
	50%	4.18 ± 0.56	1.93	6.01 ± 0.70	-1.09

Table 2. Results of missing data treatment. Missing data inserted into the attributes "Uniformity of Cell Size", "Uniformity of Cell Shape" and "Bare Nuclei" (a); "Uniformity of Cell Size" and "Uniformity of Cell Shape" (b); and "Uniformity of Cell Size" (c).

sets having 10% to 50% of missing data. Table 2-a shows that there are two occasions when there is a significant difference between the results. In the first one, the error rate of the classifier generated with 10% of missing data and without previous treatment of missing data is significantly greater than the error rate of the classifier generated with no missing data. In the second one, the error rate of the classifier induced with 30% of missing data treated by the model-based method is significantly greater than the error rate of the classifier generated with no missing data. As shown in Table 2-b, the error rates obtained with no previous treatment of missing data and with 20%, 30%, 40% and 50% of missing data were significantly greater than the error rate obtained with 0% of missing data. On the other hand, only the error rate obtained by the classifier generated with a training set with 50% of missing data is significantly greater then the error rate obtained with 0% of missing data. Finally, the results presented in Table 2-c do not shown significant differences among the classifiers error rates.

5 Conclusion

The main objective of this work is to present some initial results of a research that aims to analyse the benefits and drawbacks of missing data treatment methods [3]. In this work, we analyse the behaviour of two methods for missing data treatment: a model-based method using the k-nearest neighbour algorithm; and the internal algorithm used by C5.0 to treat missing data. Both methods were tested inserting different percentages of missing data into different attributes. The results are very promising. Both methods provide very good results, even

when the training sets had a large amount of missing data. In future works, the missing data treatment methods will be analyzed in other datasets. Furthermore, in this work missing data is being randomly inserted. In a future work, we will analyze the behaviour of both methods when missing data is not randomly distributed. In this case, there exist the possibility of invalid knowledge be created by the inducer. For an effective analysis, we will have to inspect not only the error rate, but also, the quality of the knowledge induced by the learning system.

Acknowledgements. This research is partially supported by Brazilian Research Councils CAPES and FINEP.

References

[1] J. A. Baranauskas and M. C. Monard. Reviewing some Machine Learning Concepts and Methods. Technical Report 102, ICMC-USP, São Carlos, SP, Feb 2000. ftp://ftp.icmc.sc.usp.br/pub/BIBLIOTECA/rel_tec/Rt_102.ps.zip.

[2] G. E. A. P. A. Batista and M. C. Monard. AMPSAM: Um ambiente computacional para medir a performance de sistemas de Aprendizado de máquina. In *Anais do I Encontro Nacional de Inteligência Artificial - ENIA 97*, pages 41–45, Ago 1997.

[3] Gustavo E. A. P. A. Batista. Data Pre-processing for Supervised Learning (in Portuguese). Phd Qualifying Exam, ICMC-USP, March 2000.

[4] R. Kohavi and C. Kunz. Option Decision Trees with Majority Votes. In *Proceedings of 14th International Conference in Machine Learning*, pages 161–169, San Francisco, CA, 1997. Morgan Kaufmann.

[5] K. Lakshminarayan, S. A. Harp, and T. Samad. Imputation of Missing Data in Industrial Databases. *Applied Intelligence*, 11:259–275, 1999.

[6] C. J. Merz and P. M. Murphy. UCI Repository of Machine Learning Datasets, 1998. http://www.ics.uci.edu/ mlearn/MLRepository.html.

[7] J. R. Quinlan. *C4.5 Programs for Machine Learning*. Morgan Kaufmann, CA, 1988.

[8] L. Saitta, A. Giordana, and F. Neri. What is Real World. In D. Aha and P.J. Riddle, editors, *Working Notes for Applying Machine Learning in Pratice: A Workshop of the Twelfth International Machine Learning Conference (Technical Report AIC-95-023)*, pages 43–40, Washington, DC, 1995. Navy Center for Applied Research in Artificial Intelligence.

[9] B. G. Tabachnick and L. S. Fidell. *Using Multivatiate Statistics*. Haper Collins College Publishers, 1996.

[10] D. R. Wilson and T. R. Martinez. Reduction Techniques for Exemplar-Based Learning Algorithms. *Machine Learning*, 38(3):257–286, March 2000.