Preprocessing of Missing Values Using Robust Association Rules

Arnaud Ragel

ragel@info.unicaen.fr
GREYC - CNRS UPRESA 6072 - Université de Caen
14032 Caen cedex - France

Abstract. Many of analysis tasks have to deal with missing values and have developed specific and internal treatments to guess them. In this paper we present an external method for this problem to improve performances of completion and especially declarativity and interactions with the user. Such qualities will allow to use it for the data cleaning step of the KDD[1] process[6]. The core of this method, called MVC (Missing Values Completion), is the RAR[2] algorithm that we have proposed in [14]. This algorithm extends the concept of association rules[1] for databases with multiple missing values. It allows MVC to be an efficient preprocessing method: in our experiments with the c4.5[12] decision tree program, MVC has permitted to divide, up to two, the error rate in classification, independently of a significant gain of declarativity.
keywords: Association rules, Missing Values, Preprocessing, Decision Trees.

1 Introduction

The missing values problem is an old one for analysis tasks[8] [11]. The waste of data which can result from casewise deletion of missing values, obliges to propose alternatives approaches. A current one is to try to determine these values [9]. However, techniques to guess the missing values must be efficient, otherwise the completion introduces noise. With the emergence of KDD for industrial databases, where missing values are inevitable, this problem has become a priority task [6] also requiring declarativity and interactivity during treatments. At the present time, treatments are often specific and internal to the methods, and do not offer such qualities. Consequently the missing values problem is still a challenging task of the KDD research agenda [6].

We have proposed in [14] the RAR algorithm to correct the weakness of usual association rules algorithms[2] in mining databases with multiple missing values. The efficiency of this algorithm to extract quickly all the associations contained in such a database, allows to use it for the missing values problem. That is what

[1] Knowledge Discovery in Databases
[2] Robust Association Rules

we propose in this paper with the MVC (Missing Values Completion) method. It uses the association rules to fill the missing values, independently of any analysis tasks, during a preprocessing, declarative and interactive process with the user. MVC is interesting for its declarative concept and also for its performances which allow to use it to replace the ad-hoc missing values processing of analysis tasks: for example, the error rate of the classification software c4.5[12] has been divided up to two in our experiments using MVC rather its own missing values processing.

In this paper, we focus on the missing values problem for decision trees but we will see that MVC can be interesting for any analysis task. Section 2 briefly reviews usual approaches to deal with missing values in decision trees and points out problems. In Sect. 3 we present the RAR algorithm[14] and show in Sect. 4 its interest for the missing values problem in MVC. In Sect. 5 we present experiments using MVC for classification tasks with the decision tree program c4.5[12] and conclude in Sect. 6, on the possibility with MVC to control the noise introduced during the completion step.

2 Usual Missing Values Treatments

As deleting data with missing values may cause a huge waste of data, missing values are often filled. Because of no standard and independent method to deal with missing values, many of programs have developed specific and internal treatments. This is the case for decision trees for which missing values are an important problem[9], [13].

A current approach is to fill the missing values with the most common values in the datasets constructed for the classification task. There are also some variants, like in c4.5[12], where a missing value is filled with several weighted values. Critics of such approaches are that the treatment is not easy to understand, especially with c4.5, and that as the value is chosen in datasets constructed for the analysis task, and not to decide the missing value, completion may be wrong [9], [13]: the subdatasets are constructed to be homogeneous for the class attribute and not for the attributes with missing values.

Another technique, proposed by Breiman et al. in [3], is to use a surrogate split to decide the missing value for an attribute. The surrogate attribute is the one with the highest correlation with the original attribute. If this method uses more information than the precedent ones, its efficacy depends on the possibility to find correlation between two attributes. Looking for more associations have been proposed by Quinlan and Shapiro in [10] where they use a decision tree approach to decide the missing values for an attribute. If S is the training set and A an attribute with missing values, the method considers the subset S', of S, with only cases where the attribute A is known. In S' the original class is regarded as another attribute while the value of attribute A becomes the class to be determined. A classification tree is built using S' for predicting the value of attribute A from the other attributes and the class. Then, this tree can be used to fill the missing values. Unfortunately it has been shown that difficulties arise

if the same case has missing values on more than one attribute[9]: during the construction of a tree to predict A, if a missing value is tested for an attribute B another tree must be constructed to predict B and so on. So this method cannot be used practically. More recently, in 1996, K. Lakshminarayan et. al proposed in [7] a similar approach where they use machine learning techniques (Autoclass[5] and c4.5[12]) to fill missing values. They avoid the problem of multiple missing values on a same data using the internal missing values treatment of c4.5 or of Autoclass. Consequently the machine learning approaches (Decision tree and Clustering) can be used only to decide missing values for only one attribute.

With this brief overview of missing values treatment for decision trees, which can be extended to other analysis methods, we see that the main difficulty is to find declarative associations between data in presence of multiple missing values. In the next section, we present a method which can be used for this task.

3 The Robust Association Rules Algorithm

The RAR algorithm, that we have proposed in [14], corrects a weakness of usual association rules algorithms: a lot of relevant rules are lost by these algorithms when missing values are embedded in databases. We briefly recall principles of association rules and we point out the main characteristics of the RAR algorithm.

An association rule[1] is an expression of the form $X \longrightarrow Y$, where X and Y are sets of items, and is evaluated by a support and a confidence. Its support is the percentage of data that contains both X and Y. Its confidence is the percentage of data that contains Y among those containing X. For example, the rule *fins=y and tail=y* $\longrightarrow$ *hairs=n* with a support of 13% and a confidence of 92% states that 13% of animals have fins, tail and no hairs and that 92% of animals with fins and tail have no hairs. The problem of mining rules is to find all rules that satisfy a user-specified minimum support and minimum confidence. Fast algorithms[2] [15] are used successfully to mine such rules in very large transaction databases. In such databases, the missing values problem do not exist in practically. In order to use the association rules concept in relational tables, where missing values are inevitable, we have proposed the RAR algorithm.

In order to avoid the collapse of fast association rules algorithms when a database has missing values, a key point of RAR is to discover rules in valid databases rather than in whole databases. A valid database (vdb) is defined for an itemset X as the largest database without missing values for this itemset.

For example, the first left dataset in the Fig. 1 is a vdb for the itemsets {X2}, {X3} and {X2, X3} because there is no missing value for X2 and X3. The second one is the vdb for itemsets {X1}, {X1,X2}, {X1,X3}, {X1,X2,X3} and the third one is a vdb for {X4}, {X2,X4}, {X3,X4}, {X2,X3,X4}.

We have implemented the RAR algorithm on the basis of the fast association rules algorithms [2] to assure an efficient computation time. Without this compatibility, the RAR algorithm would not be useful in practice and would not have permitted a deep exploration. The RAR algorithm has required a modifi-

Id	X1	X2	X3	X4
1	?	a	a	c
2	a	a	b	?
3	a	b	c	c
4	a	b	d	c
5	?	b	e	d
6	b	b	f	?
7	b	c	g	d
8	b	c	h	d

VDB 1

Id	X1	X2	X3	X4
1	Disabled			
2	a	a	b	?
3	a	b	c	c
4	a	b	d	c
5	Disabled			
6	b	b	f	?
7	b	c	g	d
8	b	c	h	d

VDB 2

Id	X1	X2	X3	X4
1	?	a	a	c
2	Disabled			
3	a	b	c	c
4	a	b	d	c
5	?	b	e	d
6	Disabled			
7	b	c	g	d
8	b	c	h	d

VDB 3

Fig. 1. RAR approach for dealing with missing values

cation of support and confidence definitions that we detail in [14].
This approach gives good results because missing values are temporarily ignored, without deleting data: if data are not all used together to compute one rule, they are all used to compute the ruleset. For example, in the Fig. 1, data 1 is not used to evaluate support of $\{X1, X3\}$ but it is used to evaluate support of $\{X2, X4\}$. We can see also that this approach works directly on the data and does not require any completion of missing values.

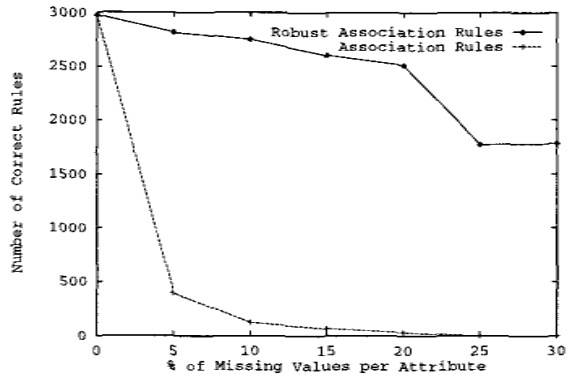

Fig. 2. Robust Association Rules Performances

Figure 2 depicts the results of an experiment comparing performances between the RAR algorithm and the main fast algorithm[2]: **a reference ruleset** is constructed from a database, 2000 data and 8 attributes, with no missing values. Then missing values are randomly introduced with a rate of 5% at first and 30% to finish with an increment of 5% on each attribute of this database. The number of correct rules, i.e include in **the reference ruleset**, retrieved with the two different approaches are shown by the curves in Fig 2. We see that the number of retrieved rules is clearly larger with RAR, especially when the number of missing values increases.

4 Interest of RAR for Missing Values

The efficiency of RAR, to find all the association rules quickly in a database with missing values, allows us to propose the method MVC. This method works as in the Fig 3, where it first extracts rules and then uses them to fill missing values, with a possible interaction with the user.

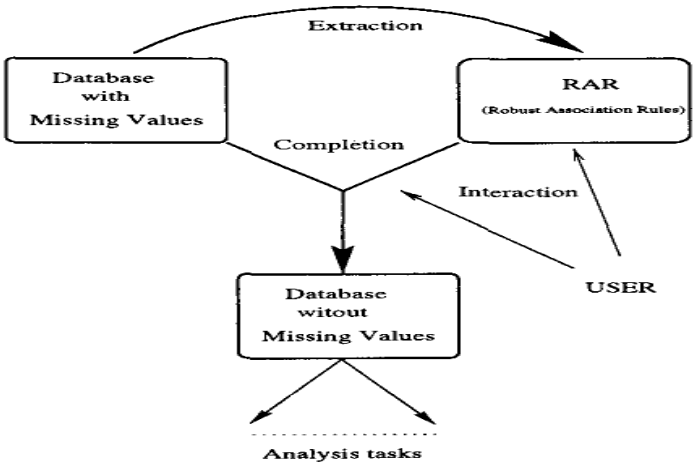

Fig. 3. The MVC Process

As we do not know which values will be missing, the first step of MVC consists of searching for all the rules above the thresholds given by the user. RAR enabled this strategy and we know that it is better, for computational time, than searching rules only when a missing value is encounted. The constructed ruleset is then used in the second step of MVC (the completion step).
For the completion, the rules matching a data and concluding on a attribute which is missing are used to decide its value. Two situations are possible:
1- All the matching rules indicate the same conclusion, then it is used.
2- Several matching rules conclude on different values. In this case the confidence measure and the number of rules concluding on a same value are used to solve the conflict automatically. Of course, the user can intervene. This approach is simple but, in our experiments, there is always a clear preference, in number of rules, for one value rather than the others.

We show below an example of completion for a data from the Zoo database[3]. In this example, *fins* and *hairs* are missing. The conclusions of the matched rules will be used to fill them:
aquatic=y, tail=y, hairs=?, legs=0, fins=?
with use of *aquatic=y, and, legs=0* $\longrightarrow$ *fins=y (sup*[4]*: 17%, conf*[5]*: 97%)*

[3] UCI Machine Learning Repository, http://www.ics.uci.edu/ mlearn/MLRepository.html
[4] support
[5] confidence

becomes: aquatic=y, tail=y, hairs=?, legs=0, fins=y
with use of *fins=y and tail=y* $\longrightarrow$ *hairs=n (sup: 13%, conf: 92%)*
becomes: aquatic=y, tail=y, hairs=n, legs=0, fins=y
Let us notice that recursion is permitted in MVC only with rules which have a high confidence (above 95%) to avoid series of wrong completion.
With this intuitive treatment the user can easily intervene in the completion process: he can visualize the ruleset, remove, change or add new rules. Furthermore the confidence of a rule can be a good indication of the correctness of the treatment: the rule with a confidence of 97%, in the previous example, indicates that 3% of noise may be introduced. It is always possible to find matched rules, to complete, if we allow the use of rules with a low support and a low confidence. This point can appear as a weakness of our approach because a low confidence may implicate noise in data. As a matter of fact, this critic can apply to any method which completes the missing values, like the ones presented in Sect. 2. But, as they cannot give a value like confidence to measure the noise, it does not point out.

In the next section we show with several experiments that, moreover to propose a preprocessing and declarative approach, MVC improves performances of analysis tasks.

5 Results

For this study, we use one of the most used decision tree programs, c4.5[12], which can handle missing values by sending them to each subset, with a weight proportional to the known values in the subset. For scientific exactness, to compare with the automatical missing values treatment of c4.5, MVC has been used without interaction with the user.

5.1 Introduction of multiple missing values

The aim of this experiment is to study the effect of missing values for a classification task, and to compare the ability of treatments (MVC and the internal method of c4.5) to correct the problem. To have a reference result, c4.5 is first run on an original database with no or few missing values. Then missing values are randomly introduced in the database and c4.5 is run twice. Firstly, using its own treatment to deal with missing values and secondly using MVC to preprocess the missing values. Then results are compared.

Two databases, coming from the UCI Machine Learning Repository has been used. The Vote database which has 435 data and 17 attributes and the Credit Card database which have 690 data and 15 attributes[6].

Tab. 1 gives the results of this experiment. We can see that classification error rate using MVC is lower: it can be divided up to two. Another result is

[6] some with missing values

Table 1. Average Results over ten trials with multiple missing values

Number of MV	Use of MVC	Unpruned Tree Size Tree	Errors (Train)	Errors (Test)	Pruned Tree Size Tree	Errors (Train)	Errors (Test)	Errors (Predic.)
Vote Database								
(Reference Result)		29.2	1.9%	4.8%	16	2.6%	4.4%	5.7%
5%	no	42.4	2.5%	7.8%	13.9	4.2%	6.7%	7.6%
10%	no	57.7	3.2%	9.2%	10.6	7.1%	8.3%	9.8%
15%	no	51.1	4.1%	8.0%	8.8	9.1%	10.1%	11.8%
20%	no	64.6	4.6%	9.7%	11.2	10.8%	12.9%	14.4%
5%	yes	35.5	2.5%	8.0%	6.1	4.6%	5.5%	6.5%
10%	yes	30.7	2.6%	6.2%	6.1	4.3%	5.3%	6.2%
15%	yes	37.6	2.6%	7.6%	8.8	4.5%	6.0%	6.7%
20%	yes	46.6	2.6%	8.5%	10.0	4.9%	6.7%	7.4%
Credit Card Database								
(Reference Result)		162.4	7.4%	16.5%	21.9	11.2%	13.2%	14.4%
5%	no	200.8	7.8%	16.8%	13.2	12.4%	13.5%	16.1%
10%	no	190.4	8.8%	17.9%	14.9	13.2%	13.8%	17.4%
15%	no	182.5	9.1%	18.0%	11.4	14.3%	15.1%	19.0%
20%	no	206.8	8.9%	16.8%	15.2	15.9%	17.8%	21.6%
5%	yes	141.5	7.4%	16.5%	23.3	11.3%	14.6%	14.9%
10%	yes	134.1	6.5%	13.9%	39.9	9.6%	13.8%	14.6%
15%	yes	155.1	6.3%	15.8%	22.1	11.0%	13.9%	14.3%
20%	yes	145.6	5.6%	13.9%	15.4	10.0%	12.0%	12.4%

that the more we introduce missing values in the credit card database the better are the results. Such a result may be explained by the fact that some unexpected values for data (e.g noise) may be set to unknown with this experiment and then are filled by MVC with typical ones.
MVC has completed all the missing values with an average value of correct completion of 97.3% for *Vote* and of 96.26% for *Credit Card*. Let us notice that we cannot have such information with c4.5 because of its internal and multiple completion approach.

5.2 Real world applications

In this experiment we use 2 databases, from real world, with missing values at origin:

1. OERTC database: This database is used, for classification tasks, in collaboration with the lymphome group of the O.E.R.T.C (European Organization of Research and Treatment of Cancer). We have used the H7 protocol which has 832 cases (1988-1993) and 27 attributes. Eleven of them have missing values with the following proportions: 52%, 43%, 88%, 84%, 39%, 8%, 6%, 7%, 2%, 6% and 6.5%.

2. Auto insurance database: it comes from the UCI Machine Learning Repository and has 205 cases with 25 attributes[7]. Six of them have missing values with the following proportions: 20%, 0.97%, 1.95%, 1.95%, 0.97%, 0.97%, 1.95%.

Tab. 2 gives the results. In the OERTC database results are significatively increased using MVC. An interesting point is that MVC has made to emerge an attribute not used otherwise. With it, qualities of the tree seem to be better (a predicted error of 6.8%). In the Auto database, results are the same but, the low rates of missing values and the few cases, cannot really decide if MVC could be useful. However MVC has completed missing values with an unique value, contrary to c4.5, which is better for the understanding.

Table 2. Average Results over ten trials on real world applications with c4.5

Database	Use of MVC	Unpruned Tree Size of Tree	Errors (Train)	Errors (Test)	Pruned Tree Size (of Tree)	Errors (Train)	Errors (Test)	Errors (Predic.)
OERTC	no	91.4	4.4%	8.5%	41.3	6.2%	7.3%	10.0%
	yes	89.7	2.2%	5.6%	40.4	3.5%	6.5%	6.8%
Autos	no	152.1	4.0%	16.6%	88.3	8.3%	19.5%	30.0%
	yes	149.2	4.3%	16.0%	83.4	8.3%	18.5%	29.9%

5.3 Conclusion on Results

Results seem to be noticeably improved in classification when there are many missing values. Otherwise, results are at least as good with the preprocessing. Unlike c4.5 approach, another advantage is that the completion is made with only one value which allows the user to understand and react about it.

6 Conclusion

Contrary to previous approaches reviewed in Sect. 2, the ability of RAR to discover associations in missing values databases has permitted to propose an efficient association technique. This method, MVC, uses RAR to guess the missing values in databases. Such an approach leads to a significant gain of performances: the error rate in classification, with c4.5[12], can be divided up to two, if MVC is used rather than the classical missing values treatment of c4.5. But the main advantage of MVC is to be a preprocessing method, independent of any analysis tasks, which offers a more understandable treatment of the missing values and a possible interaction with the user. Such qualities, rarely available as far as we

[7] numeric attributes have been discretized for this experiment

know for the missing values problem, may enable MVC to become a tool for the data cleaning step of the KDD process [6].

At the present time, we are working on several points to improve use of rules in MVC. One of them, is a definition of a new rule score, specially designed for the completion problem. A second one is on an adjustment of threshold completion: first experiments, using the noise measure given by the confidence of rules, seem to indicate that it is better to stop completion rather than introducing to much noise.

References

1. R. Agrawal, T. Imielinski, and A. Swami. Mining association rules between sets of items in large databases. In Proc. of the ACM SIGMOD Conference on Management of Data, Washington, D.C., p 207-216, May 1993.
2. R. Agrawal, H. Mannila, R. Srikant, H. Toivonen and A. I. Verkamo. Fast Discovery of Association Rules. In Advances in Knowledge Discovery and Data Mining, Chapter 12, AAAI/MIT Press, 1996.
3. L. Breiman, J.H Friedman, R.A Olshen, C.J Stone. Classification and Regression Trees, Wadsworth Int'l Group, Belmont, CA, The Wadsworth Statistics/Probability Series, 1984.
4. G. Celeux. Le traitement des données manquantes dans le logiciel SICLA. Technical reports number 102. INRIA, France, December 1988.
5. P. Cheeseman, J. Kelly, M. Self, J. Stutz, W. Taylor and D. Freeman. Bayesian Classification. In Proc. of American Association of Artificial Intelligence(AAAI), 607-611, San Mateo, CA, 1988.
6. U.M Fayyad, G. Piatetsky-Shapiro, and P. Smyth. From data mining to knowledge discovery: An overview. In Advances in Knowledge Discovery and Data Mining, pages 1-36, AAAI/MIT Press, 1996.
7. K. Lakshminarayan, S.A Harp, R. Goldman and T. Samad. Imputation of missing data using machine learning techniques. Proc. of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96), AAAI/MIT Press, 1996.
8. R.J.A Little, D.B Rubin. Statistical Analysis with Missing Data. John Wiley and Sons, N.Y., 1987.
9. W.Z Liu, A.P White, S.G Thompson and M.A Bramer. Techniques for Dealing with Missing Values in Classification. In Second Int'l Symposium on Intelligent Data Analysis, London, 1997.
10. J.R Quinlan. Induction of decision trees. Machine learning, 1, p. 81-106, 1986.
11. J.R Quinlan. Unknown Attribute Values in Induction, in Segre A.M.(ed.), Proc. of the Sixth Int'l Workshop on Machine Learning, Morgan Kaufmann, Los Altos, CA, p. 164-168, 1989.
12. J.R Quinlan. C4.5: Programs for Machine Learning, Morgan Kaufmann, San Mateo, CA, 1993.
13. A. Ragel: Traitement des valeurs manquantes dans les arbres de décision. Technical reports, Les cahiers du GREYC. University of Caen, France, 1997.
14. A. Ragel and B. Crémilleux. Treatment of Missing Values for Association Rules. In Proc. of The Second Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD-98), p. 258-270, Melbourne, Australia, 1998.
15. H. Toivonen. Sampling large databases for association rules. In Proc. of the 22nd Int'l Conference on Very Large Databases (VLDB'96), p. 134-145, India, 1996