Efficient Missing Data Imputation for Supervised Learning

Shichao Zhang [1], Xindong Wu [2,3], Manlong Zhu [4]

[1] Department of Computer Science, Zhejiang Normal University, China
[2] School of Computer Science and Information Technology, Hefei University of Technology, China
[3] Department of Computer Science, University of Vermont, USA
[4] College of Computer Science and Information Technology, Guangxi Normal University, China
zhangsc@zjnu.cn; xwu@cems.uvm.edu; rpzml@163.com

Abstract

In supervised learning, missing values usually appear in the training set. The missing values in a dataset may generate bias, affecting the quality of the supervised learning process or the performance of classification algorithms. These imply that a reliable method for dealing with missing values is necessary. In this paper, we analyze the difference between iterative imputation of missing values and single imputation in real-world applications. We propose an EM-style iterative imputation method, in which each missing attribute-value is iteratively filled using a predictor constructed from the known values and predicted values of the missing attribute-values from the previous iterations. Meanwhile, we demonstrate that it is reasonable to consider the imputation ordering for patching up multiple missing attribute values, and therefore introduce a method for imputation ordering. We experimentally show that our approach significantly outperforms some standard machine learning methods for handling missing values in classification tasks.

Index Terms — *Artificial intelligence*; *Missing data imputation*; *Data processing*.

1. Introduction

Many real world databases are incomplete as some instances may have missing attribute-values. Attribute values can be missing for various reasons. It may be due to a malfunction of equipment, absence of a measuring unit and precision, conversion between non-compatible entities, or erroneous human imputation. Besides, data might be missing because not enough information has been collected from their original sources. This presents a problem in data analysis as many machine-learning algorithms are based on the assumption that the data are complete. With this incompleteness, analysts interested in using the data for parameter estimation or statistical inference are handicapped because most common data analytic packages analyze only complete cases. These imply that a reliable method for dealing with missing values is necessary.

Missing values are an unavoidable problem in dealing with most of the real world data sources, and various methods for dealing with such data have been developed, particularly in the context of missing data in sample surveys. Imputation is a popular strategy in comparison with other methods, such as deleting the instances containing missing values. Missing data imputation is a procedure that replaces the missing values in a dataset by some plausible values. One advantage of this approach is that the missing data treatment is independent of the learning algorithm used. This allows users to select the most suitable imputation method for their applications. Commonly used imputation methods for missing response values include parametric and non-parametric regression imputations.

Many missing data analysis techniques are of single-imputation, such as decision tree imputation, non-parametric imputation method, parametric imputation method, and so on. Recently, much research on missing data analysis has focused on multi-imputation techniques [1-5] or iterative imputation methods [3,6] for addressing the issues in single-imputation.

In this paper, we show that it is necessary to iteratively impute multiple missing values rather than single imputation in real-world applications. We propose an EM-style iterative imputation method, in which each missing attribute-value is iteratively filled using a predictor constructed from the known values and predicted values of the missing attribute-values from the previous iterations. On the other hand, we demonstrate that it is reasonable to consider the imputation ordering for patching up multiple missing attribute values and propose a method for imputation ordering. We experimentally show that our

Proc. 9th IEEE Int. Conf. on Cognitive Informatics (ICCI'10)
F. Sun, Y. Wang, J. Lu, B. Zhang, W. Kinsner & L.A. Zadeh (Eds.)
978-1-4244-8040-1/10/$26.00 ©2010 IEEE

approach significantly outperforms some standard machine learning methods for handling missing values in classification tasks.

The rest of this paper is organized as follows. Section 2 briefly reviews related work on missing attribute value imputation and imputation ordering. In Section 3, we design a principle of imputation ordering to predict missing attribute values. Section 4 describes our experiments on UCI datasets. We conclude this paper in Section 5.

2. Related work

Currently, there are two mainstream directions for dealing with the missing value completion (imputation) problem. One is based on machine learning, including auto associative neural networks, decision tree imputation, and so on. Another is based on statistics, including linear regression, multiple imputation, parametric imputation, and non-parametric imputation.

With single-imputation techniques, missing values in a variable are filled in by a plausible estimate such as the mean or median for that variable on other participants. Better estimates may be obtained by a regression model on expected values, or a hot deck procedure. However, single-imputation cannot provide valid standard errors and confidence intervals, since it ignores the uncertainty implicit in the fact that the imputed values are not the actual values. On the other hand, imputing a single value does not capture the sample variability of the imputed value, or the uncertainty associated with the model used for imputation.

The main idea of multiple-imputation [3] is to fill in missing values by drawing from the posterior predictive distribution of the missing data given the observed data. The procedure is independently repeated M times, so it cannot efficiently utilize all the observed values including the observed ones whose instances contain missing values. The EM algorithm [6] repeatedly alternates between two steps, the E step and the M step. All the two steps depend on parametric models, and the later imputation will relate to the result of the former. If the model is mis-specified (in fact, in real-world applications, it is usually impossible for us to know the distribution of the real dataset), the estimations of the parametric method may be highly biased and optimal control factor settings may be miscalculated. So it is unreasonable to use a parametric model to deal with the problem and the non-parametric method can provide a better alteration. In this paper, we employ a non-parametric method (the *kNN* algorithm) to impute missing values.

Obviously, multiple imputation including MI and EM algorithm is computationally more expensive compared to single imputation procedures, especially for the EM algorithm, as the user cannot predict the number of iterations for convergence. [7] presented an iterative non-parametric algorithm which is similar to the EM algorithm, and an interesting feature of the algorithm is that the E and M steps collapse into a single step because the data being filled in are the modal-updating and the filled-in values and update the model at the same time. The algorithm can utilize essentially all observed values including the observed ones whose instances contain missing values. [7] also demonstrated experimentally that the speed of convergence is faster than the EM algorithm. But they cannot pay attention to the imputation order during the process of imputing multiple missing values.

In fact, it is very important for us to consider the imputation order during the process of imputing multiple missing values. [8-10] argued that it can be efficient to improve the prediction accuracy and decrease the classification error rate if we can apply an appropriate imputation order during imputing. [10] dramatically improved the efficiency of imputation with an ordering process of an exhaustive search. However, it is extremely expensive in time complexity if the dataset contains too many attributes. For example, when there are *n* attributes, there are *n!* different possible orderings. [8] presented a method to impute multiple missing values, which uses a lexicographic ordering and an iterative imputation method to impute the missing values. There are at least two disadvantages: firstly, the discretization of the continuous values will lose useful information. Second, it is not substantial for the algorithm to consider the significance between the attribute and class label by mutual information. However, in practice, the ratio of missing values in comparison to the observed values in one instance or in one column in the dataset will have an impact for the performance of imputation especially for iterative imputation methods. As these methods use imputation results from the previous iterations to impute the missing values in the current iteration, it will have a significant impact for us to impute missing values with these methods if the previous results contain a serious bias.

In this paper, we present a novel imputation ordering strategy which makes a trade-off between the impact of the missing rate in one instance and the impact of the relation between the attribute and the class label in an EM-like iterative imputation method, but it is different from the MI and EM algorithms. In the first iteration imputation, we use the mean (or mode) values of all the observed attribute values to fill in the missing values in order to make the best use of all the information. From the second iteration of the imputation process, iterative imputation is based on the results of the previous imputation, then we present a principle of imputation ordering, which is employed a harmonic mean allowing users to specify their own desired trade-off in terms of the impact of the missing value ratio and the impact of the relation between the attribute and class label, to impute missing values in order. This procedure is stopped when the average change of imputed values is approximately stabilized, or satisfies a given requirement by the user. The non-parametric imputation [11] is utilized in this algorithm because we cannot exactly know the model of data, because we have usually not priori knowledge about the data.

3. Our Algorithm

In this section, we first demonstrate that we need to iteratively impute multiple missing values in order to make the best use of the observed values in Section 3.1, and then explain in detail in Section 3.2 an imputation ordering for multiple missing values with respect to the impact of the ratio of missing values in one attribute and the relation between the conditional attributes and class label.

3.1 Making the best use of the observed values

Most imputation methods try to impute missing values using those instances whose attribute values are all observed. For example, non-parametric imputation methods, such as the kernel approach, impute the missing values through utilizing those instances that don't have any missing values as the reference instances during the training process. These methods may possibly ignore two facts. First, they are normally for datasets that contain missing values. For example, in Table 1, we analyze 6 datasets from UCI [13] as follows, and there are very small percentages of missing values in these datasets. The percentage of missing values in Water-Treatment, Hepatitis, Bridge, Echocardiogram, Soybean and House-Voting is only 2.95%, 5.67%, 5.56%, 7.69%, 6.63% and 4.13% respectively, but the percentage of instances with missing values reaches 26.56%, 48.38%, 35.18%, 53.79%, 13.36% and 46.68% respectively. The OI/NC (which denotes the mean of the observed instances for each class) is only 29, 40, 11, 30, 14 and 100 respectively and the number of OI/NC must be beyond 30 for a non-parametric imputation method in a large sample in statistics. In practice, most industrial databases have a more serious problem with missing values. Take [12] as an example. Of the 4383 records in this database, none of the records are complete and only 33 out of the 82 variables have more than 50% values. Second, an incomplete instance may already contain enough information for model construction, even though it still contains missing values. For example, in practice, the values of all instances with the same class label may be missing on a particular value because of a especial reason, such as, the value of attribute 'age' is usually left empty in questionnaires because women might not want their real ages to be known. In Table 2, the values on the 5th attribute are missing and their class label is '1', but we can assume that the missing values are known. On the other hand, it is unreasonable for us to find the nearest neighbor among the instances without missing values (such as instances *d* and *e*) because their class label is '0', while the class label is '1' for instance *a* (or *b* or *c*).

So it is reasonable for us to impute missing values with instances that have observed values including those instances which contain some missing values based on the above analysis. However, we cannot calculate the relation between the instances whose values are all observed and the instances with missing values because existing methods try to deal with all the instances without missing values. Caruana in [7] designed an iterative imputation method to resolve this problem, and in this paper, we construct iterative imputation methods to utilize all observed values.

The notations for different columns in Table 1 are as follows. M/O stands for the ratio of the number of missing attributes to the number of all attributes in a dataset; M/A stands for the ratio of the number of missing instances to the number of all instances in the dataset; M/AM stands for the ratio of the number of missing instance to the number of instances with missing instances; Multi denotes the maximal number of missing values in an instance; NC denotes the number of classes; and MR denotes the missing rate of the dataset.

Table 1: Examples in UCI Datasets

	M/O	M/A	M/AM	Multi	NC	MR
Water	31/38	26.56%	115/140	15	13	2.95%
Hepatitis	15/20	48.38%	37/75	14	2	5.67%
Bridge	9/13	35.18%	21/38	6	6	5.56%
Echocar	12/13	53.79%	39/71	9	2	7.69%
Soybean	34/35	13.36%	41/41	30	19	6.63%
House	15/16	46.68%	86/203	16	2	4.13%

Table 2. '-'denotes an observed value and '?'denotes a missing value in a database

	C_1	C_2	C_3	C_4	C_5	D
a	?	-	?	?	?	1
b	-	?	-	-	?	1
c	?	-	-	-	?	1
d	-	-	-	-	-	0
e	-	-	-	-	-	0

3.2 Imputation Order

In this paper, we employ a non-parametric method [11] which is based on an instance imputation method to impute multiple missing values. In most cases, existing algorithms are applied without special attention to the given order of missing values. However, [9] argued that it is necessary to design an imputation ordering when there are multiple missing values in one instance. For example, in Table 2, there are two missing values in instance *b*, and we must choose one of them to impute at first. So we must adopt a criterion for imputation ordering. Usually, there are many factors that have an impact on the imputation order, such as the relation between the attribute and the class label, the missing percentage in a row/column, and so on. There are many methods for imputation ordering, such as [8-10]. In this paper, we present a new and efficient imputation ordering method to deal with multiple missing values.

Based on the principle of choosing the most informative attributes to impute the missing values and the fact that an imputation algorithm is actually an instance-based learning algorithm, we consider the impact of the missing rate in each instance and the impact of the mutual information (MI) between each attribute and the class label to decide the current imputable missing value (CIMV for short). The more observed information available, we believe, the lower missing rate an attribute is. We also consider that an attribute is more important when its MI is larger. Given a dataset with missing values, we first calculate the percentage of the number of missing values in an instance, and take the inverse of this value as the impact weight for the instance (denoted as R_i: the impact weight of the ith instance), then we compute the mutual information between each attribute and the class label with all observed instances and regard mutual information as W_i (W_i: the mutual information between the ith attribute and the class label). At last, we integrate R_i and W_i as significance (denoted as *Sign*(*i,j*): the significance of the missing value which is located in the ith instance and on the jth attribute). We use the F-measure which is commonly used and was firstly introduced in information retrieval and natural language processing communities to express *Sign*(*i,j*). The F-measure requires us to specify a desired trade-off between R_i and W_i through a variable *Sign*(*i,j*). That is to say, using the F-measure allows users to specify their own desired trade-off in terms of R_i and W_i. In fact, the F-measure is a harmonic mean of R_i and W_i. Mathematically, *Sign*(*i,j*) is defined as

$$\mathrm{Sign(i,j)} = \frac{1}{2}\left(\frac{1}{W_j} + \frac{1}{R_i}\right) \qquad (1)$$

where *i* is an instance and *j* is an attributes in a given dataset.

Assuming R_i has a weight of $\alpha \in (0, +\infty)$, W_i has a weight of 1, and α is decided by the user, then the weighted harmonic mean of R_i and W_i is

$$Sign(i, j) = \frac{(\alpha + 1)R_i W_j}{R_i + \alpha W_j} \qquad (2)$$

where *i* is an instance and *j* is an attributes in the dataset.

With Eq. 2, we can rank all missing values by the *Sign*(*i,j*) values (in ascending order), and select the missing value with the least *Sign*(*i,j*) values to impute. In our method, the missing value with a larger W_i and a smaller R_i always has a priority to be selected as *CIMV*. In fact, we usually select the missing value which is the only missing value in an instance to be imputed at first. This will result in less bias. At the same time, we will select the missing value with a smaller R_i to impute when multiple missing values have the same *Sign*(*i,j*) value. Our experiments demonstrate that the results will get better if we can select an appropriate α value in different datasets.

During the first iteration the missing values have not yet been patched up. Previous efforts, for instance, kernel methods and decision trees, only regard the instances without missing values as examples of the training set. This will waste all the information of the observed values whose instances contain missing values as well, and it is unreasonable to impute missing values by utilizing all the observed instances in real-world applications. In this paper, we apply the first imputation strategy to make the best use of all the information of observed values, and we don't stop iteratively imputing the missing values using the known values and predicted values of the missing attribute-values from the previous iterations until the algorithm converges. We compute the mean when an attribute with missing values is continuous, and compute the mode if the attribute is discrete or symbolic, from the instances whose values are all observed. I.e. we use the mean (or mode) as the initial filled-in value for each missing value. Using the attribute mean (or mode) to replace missing values is a popular imputation method in machine learning and statistics. [14] argued that to calculate the mean (or mode) only from observed values is valid if and only if the dataset is chosen from a population with a normal distribution, and it is, however, usually impossible in real world applications to assume this normal distribution because we cannot know in advance the real distribution of a dataset. So performing iterative imputation for missing values is reasonable based on the previous imputation. Our algorithm is presented in Figure 1 as follows.

```
Initialization:   //Get a complete dataset after the first
imputation
    For each missing value in the given dataset
      labeling : $literation$=1, $impute$=0
      Calculate $Sign(i,j)$
    Fill with the mean (for a continuous attribute) or
mode (for a nominal attribute).
    Sort all $Sign(i,j)$
    Rank $CIMV(i)$ in ascending order
```

```
Repeat the following two steps until convergence (k iterations).
    For i=1 to (Number of missing values)
    Impute CIMV(i) utilizing all the dataset based on the kNN algorithm
    literation ++;
    impute=1;

Output:
Results with filled-in values for all missing values.
```

Figure 1: Pseudo-code of the proposed algorithm

4. Experimental Study

This section describes our experiments on some UCI datasets [13] which are conducted to demonstrate the performance of our algorithm. In Section 4.1, we present the convergence of the filled-in values, and the prediction accuracy for continuous missing values are shown in Section 4.2. We demonstrate classification errors in Section 4.3.

4.1 Convergence of the imputed values

Each iteration of the EM algorithm is guaranteed to be non-decreasing in maximum likelihood [15], thus EM converges to a local maximum in likelihood. In our algorithm, the first iteration, which uses the mean (or mode) as the initial filled-in value of for each missing value with complete instances of the same class label, obviously converges. But in the process of other iterations, we are not able to provide a similar analysis for a non-parametric method, and the reason is that there are few theoretical results regarding the validity of *kNN* in the literature, because it is difficult to build a mathematical proof. In this section we empirically show the convergence of the *kNN* method when applied to the UCI datasets.

An imputation method has converged if the "mean change in filled-in values" drops to zero. [7] argued that the "mean change in filled-in values" does not drop all the way to zero and only trends to a value which trends fast and stably to zero in non-parametric models (such as *k*NN and kernel methods).

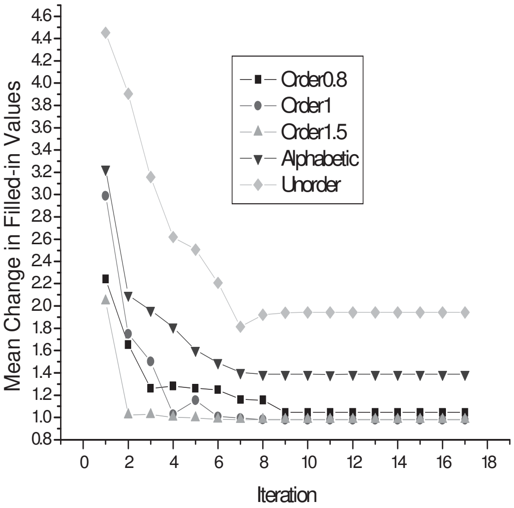

Fig. 1 *The average change in filled-in values for dataset chocardiogram*

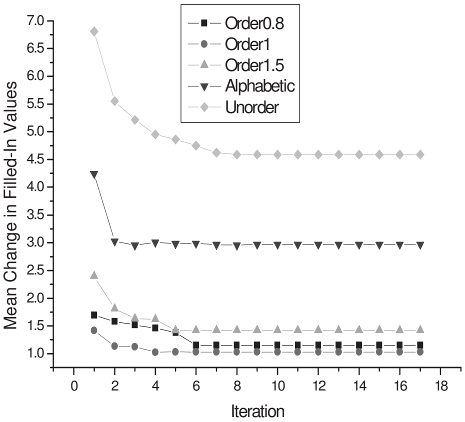

Fig.2: *The average change in filled-in values for dataset Water-Treatment*

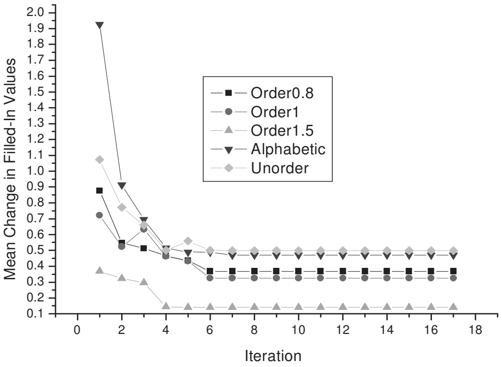

Fig. 3: *The average change in filled-in values for dataset Hepatitis Diagnosis*

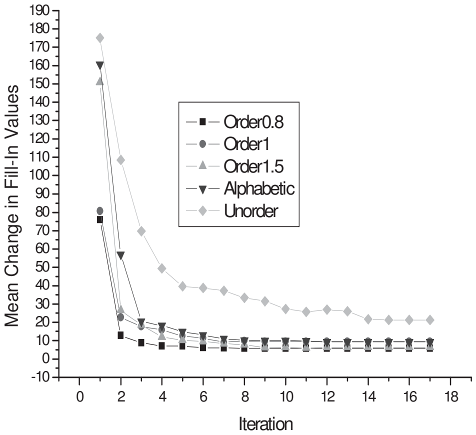

Fig.4: *The average change in filled-in values for dataset Soybean*

In Figures 1 to 4, we present the average change in filled-in values for datasets “Echocardiogram”, “Water-Treatment”, “the Hepatitis Diagnosis Problem”, and “Soybean” for successive iterations. Attribute values were normalized with variance 1.0 and the optimal *k* is selected with different *k* values in the *kNN* algorithm.

4.2 Experimental Evaluation on Prediction Accuracy

The prediction accuracy is evaluated on the Iris dataset, which contains 150 instances (50 in each of the three classes), 4 numerical attributes and no missing values in the dataset in order to demonstrate the approach’s effectiveness.

In the experiments, each instance was preprocessed to be normalized with variance 1.0 and each attribute took values between zero and one. Furthermore, we iteratively impute the missing values at different times with different algorithms in order to make sure that all algorithms can converge, and only the results of the optimal k are presented in Table 3 corresponding to the best performance due to space constraints. We inject missing values at random on different attributes and the missing rate is fixed to 1%, 5%, 10%, 20% and 30% respectively. 10-fold cross-validation is adopted in each experiment, and the accuracy of prediction is measured using the Root Mean Square Error (RMSE), as follows:

$$RMSE = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(e_i - \tilde{e}_i)^2} \quad . \qquad (3)$$

where e_i is the original attribute value; $\tilde{e}_i$ is the estimated attribute value, and m is the total number of predictions. The more RMSE is, the less the prediction accuracy is.

Also, our proposed approach is evaluated on the *Pima* dataset, which contains 768 instances with 8 numerical attributes (500 instances for class ‘0’ and 268 for class ‘1’) and there are no missing values in the dataset. Each instance was preprocessed to be normalized with variance 1.0 and each attribute takes a value between zero and one. Furthermore, we iteratively impute the missing values at different times with different algorithms in order to make sure that all algorithms can converge, and only the results of the optimal k are presented in Table 4 corresponding to the best performance due to space constraints. The missing values are injected at random on different attributes and the missing rate is fixed to 1%, 5%, 10%, 20% and 30% respectively.

We can make the following observations based on Tables 3 and 4:

1. The algorithms with imputation ordering can significantly outperform the algorithm without imputation ordering. The performance of all different α values is better than the lexicographic algorithm with regard to the prediction accuracy.
2. Exhaustive search is the optimal algorithm and all the values are the best, but the time complexity is exponential. For example, there are 8 attributes in dataset *Pima*, and therefore there are 8!= 40320 different orders for this algorithm. At the same time, the results of our algorithm are not the best, but we can get the imputation values which are not significantly different from the best values when we can select an optimal α value. For example, in Table 3, the imputation values of the exhaustive search algorithm are 0.000125, 0.00043, 0.000821, 0.00152 and 0.01881 respectively in different missing rates, and the best values of our algorithm are 0.000128, 0.000523, 0.000891, 0.001709 and 0.01879 respectively.
3. The prediction accuracy is higher when the value of α is less than 1 in different missing rates in the Iris dataset. However, the situation is inversed in Table 4. In our experiments, we have found that most of the mutual information between an attribute and the class label is less than 1 in the Iris dataset and most of the values are larger than 1 in dataset Pima. So perhaps the weight of the missing rate increases in Iris and decrease in Pima.

Table 3: *Experimental results on the Iris dataset for four algorithms (the missing rate is 5%, 10%, 20% and 30% respectively, and the algorithms include our algorithm with 5 different* α *values, lexicographic /alphabetic, and no ordering imputation)*

	5%	10%	20%	30%
Order(0.8)	0.0009	0.0010	0.0020	0.0256
Order(1)	0.0006	0.0011	0.0026	0.0325
Order(1.5)	0.0005	0.0014	0.0031	0.0986
Alphabetic	0.004	0.0082	0.0120	0.2015
Un-order	0.5424	0.7533	1.0254	1.9563

Table 4: *Experimental resulst on the Iris dataset for four algorithms*

	5%	10%	20%	30%
Order(0.8)	0.8581	1.3785	1.5795	1.85656
Order(1)	0.7666	1.2660	1.4182	1.44388
Order(1.5)	0.7125	1.2035	1.3268	1.35611
Alphabetic	3.6235	5.2351	9.2155	15.3707
Un-order	9.1046	11.017	26.672	39.1283

4.3 Evaluation on Classification Error Rate

Six UCI datasets (i.e., Echocardiogram, Water-Treatment, the Hepatitis Diagnosis Problem, Soybean, House-Voting and Bridge) are applied to compare the performances of five algorithms.

Table 5 shows that the classification error rates of the three algorithms are similar to the results of convergence as shown before. Based on the results of Table 5 and the mutual information between each conditional attribute and the class label, we can make a conclusion that the best α value is less than 1 in the datasets of Bridge, Echocardiogram and Soybean, and the best α value in the datasets of Hepatitis, House and Water-Treatment is larger than 1.

Table 5: *The Classification Error Rates of three imputation methods in different six datasets (" O_(0.8)"represents order(0.8), "Alph" is* Alphabetic*, and "Un_O"is unorder)*

	O_(0.8)	O_(1)	O_(1.5)	Alph	Un-O
Bridge	0.1731	0.1703	0.1602	0.1806	0.2305
Hepatitis	0.1703	0.1852	0.2350	0.2517	0.2758
House	0.2015	0.2103	0.2196	0.2352	0.2864
Echocar	0.2127	0.2074	0.2145	0.2258	0.2985
Water	0.2197	0.2859	0.2946	0.2625	0.3033
Soybean	0.2869	0.2510	0.2567	0.2800	0.3050

5. Conclusions

In supervised learning, missing attribute values can usually appear in the training set. These missing values may generate bias, affecting the quality of the supervised learning process or the performance of classification algorithms. Existing imputation algorithms are usually based on single imputation, which cannot provide valid standard errors and confidence intervals since they ignore the uncertainty implicit in the fact that the imputed values are not the actual values.

This paper has proposed a strategy for imputation ordering with iterative imputation for multiple missing values. At first, we advocated that it is reasonable to impute missing values with all observed values in a dataset in practice, and it is it is essential for the user to iteratively impute multiple missing values in order to make the best use of all the observed values including that observed ones whose instances contain missing values. Then the paper presented a strategy of imputation ordering which combines and trades-off the impact of the missing rate and the impact of mutual information between the conditional attributes and class label. At last, our empirical results demonstrated that the proposed method is better than the lexicographic ordering imputation method and no ordering imputation order, in terms of the number of iterations for convergence, prediction accuracy and classification error rate. The paper also showed experimentally that different α values have an impact on the performance of the algorithm. Our future work will include an analytical study on the impacts of different α values.

6. Acknowledgements

This work has been supported by the Australian Research Council (ARC) under grant DP0985456, the National Natural Science Foundation of China (NSFC) under grants 90718020 and 60828005, the 973 Program of China under award 2009CB326203, the U.S. National Science Foundation (NSF) under award CCF-0905337, the MOE Project of Key Research Institute of Humanities and Social Sciences at Universities (07JJD720044), and the Guangxi NSF (Key) grants.

References

[1] Faris, P., et al. (2002). Multiple imputation versus data enhancement for dealing with missing data in observational health care outcome analyses. *Journal of Clinical Epidemiology,* 55: 184–191.

[2] Little, R. & Rubin, D. (1987). *Statistical Analysis with Missing Data*. New York: John Wiley & Sons, Inc.

[3] Scheffer, J. (2002). Dealing with Missing Data. *Res. Lett. Inf. Math. Sci.,* 3, 153-160.

[4] Taylor, J., Murray, S. and Hsu C. (2002): Survival estimation and testing via multiple imputation. *Statistics & Probability*, 58: 221-232.

[5] Zhang, L. (2004). Nonparametric Markov chain bootstrap for multiple imputation. *Computational Statistics & Data Analysis*, 45(2): 343-353.

[6] Dempster, A.P., Laird, N.M. and Rubin, D.B., Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, series B, Vol. 39, pp. 1–38, 1977.

[7] Caruana, R. A Non-parametric EM-style algorithm for Imputing Missing Value. *Artificial Intelligence and Statistics*, January 2001.

[8] Conversano, Claudio (2003), Incremental Tree-Based Missing Data Imputation with Lexographic Ordering, *Computing Science and Statistics*, 35, 2003.

[9] NUMAO et al. (1999).Ordered estimation of missing values. Lecture notes in computer science, the Proceedings of PAKDD'99,499-503.

[10] Estevam R.H., et al. (2006), Bayesian network for imputation in classification problems. *Journal of Intelligent Information Systems*, DOI 10.1007/s 10844 –006 –0016 -x.

[11] Cover, T.M. and Hart, P.E. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, Vol. 13, No. 1, pp. 21–27, 1967.

[12] Lakshminarayan K., et al. (1999), Imputation of missing data in industrial databases, Applied Intelligence 11, 259-275.

[13] Blake, C. and Merz, C. UCI Repository of machine learning databases.1998.

[14] Brown, M.L. Data mining and the impact of missing data. *Industrial Management & Data Systems*, 103/8, 611-621, 2003.

[15] Dempster, A.P., Laird, N.M. and Rubin, D.B., Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, Series B, Vol. 39, pp. 1–38, 1977.