Missing Value Imputation Based on Data Clustering*

Shichao Zhang[1], Jilian Zhang [2], Xiaofeng Zhu[1], Yongsong Qin[1], and Chengqi Zhang[3]

[1] Department of Computer Science, Guangxi Normal University, Guilin, China
[2] School of Information Systems, Singapore Management University, Singapore
[3] Faculty of Information Technology, University of Technology Sydney
P.O. Box 123, Broadway, NSW2007, Australia
`zhangsc@mailbox.gxnu.edu.cn, zhangjilian@yeah.net,`
`xfzhu_dm@163.com, ysqin@mailbox.gxnu.edu.cn,`
`chengqi@it.uts.edu.au`

Abstract. We propose an efficient nonparametric missing value imputation method based on clustering, called CMI (Clustering-based Missing value Imputation), for dealing with missing values in target attributes. In our approach, we impute the missing values of an instance *A* with plausible values that are generated from the data in the instances which do not contain missing values and are most similar to the instance *A* using a kernel-based method. Specifically, we first divide the dataset (including the instances with missing values) into clusters. Next, missing values of an instance *A* are patched up with the plausible values generated from *A*'s cluster. Extensive experiments show the effectiveness of the proposed method in missing value imputation task.

1 Introduction

Missing values imputation is an actual yet challenging issue confronted in machine learning and data mining [1, 2]. Missing values may generate bias and affect the quality of the supervised learning process or the performance of classification algorithms [3, 4]. However, most learning algorithms are not well adapted to some application domains due to the difficulty with missing values (for example, Web applications) as most existed algorithms are designed under the assumption that there are no missing values in datasets. That implies that a reliable method for dealing with those missing values is necessary. Generally, dealing with missing values means to find an approach that can fill them and maintain (or approximate as closely as possible) the original distribution of the data. For example, in a database, if the known values for an attribute *A* are: 2 in 60% of cases, 6 in 20% of cases and 10 in 10% of

---

cases, it is reasonable to expect that missing values of A will be filled with 2 (if A is discrete) or 3.4 (if A is continuous) (see [5]).

Missing values may appear either in conditional attributes or in class attribute (target attribute). There are many approaches to deal with missing values described in [6], for instance: (a) Ignore objects containing missing values; (b) Fill the missing value manually; (c) Substitute the missing values by a global constant or the mean of the objects; (d) Get the most probable value to fill in the missing values. The first approach usually lost too much useful information, whereas the second one is time-consuming and expensive in cost, so it is infeasible in many applications. The third approach assumes that all missing values are with the same value, probably leading to considerable distortions in data distribution. However, Han et al. 2000, Zhang et al. 2005 in [2, 6] think: 'The method of imputation, however, is a popular strategy. In comparison to other methods, it uses as more information as possible from the observed data to predict missing values.

Traditional missing value imputation techniques can be roughly classified into parametric imputation (e.g., the linear regression) and non-parametric imputation (e.g., non-parametric kernel-based regression method [20, 21, 22], Nearest Neighbor method [4, 6] (referred to as NN)). The parametric regression imputation is superior if a dataset can be adequately modeled parametrically, or if users can correctly specify the parametric forms for the dataset. For instance, the linear regression methods usually can treat well the continuous target attribute, which is a linear combination of the conditional attributes. However, when we don't know the actual relation between the conditional attributes and the target attribute, the performance of the linear regression for imputing missing values is very poor. In real application, if the model is misspecified (in fact, it is usually impossible for us to know the distribution of the real dataset), the estimations of parametric method may be highly biased and the optimal control factor settings may be miscalculated.

Non-parametric imputation algorithm, which can provide superior fit by capturing structure in the dataset (note that a misspecified parametric model cannot), offers a nice alternative if users have no idea on the actual distribution of a dataset. For example, the NN method is regarded as one of non-parametric techniques used to compensate for missing values in sample surveys [7]. And it has been successfully used in, for instance, U.S. Census Bureau and Canadian Census Bureau. What's more, using a non-parametric algorithm is beneficial when the form of relationship between the conditional attributes and the target attribute is not known a-priori [8].

While nonparametric imputation method is of low-efficiency, the popular NN method faces two issues: (1) each instance with missing values requires the calculation of the distances from it to all other instances in a dataset; and (2) there are only a few random chances for selecting the nearest neighbor. This paper addresses the above issues by proposing a clustering-based non-parametric regression method for dealing with the problem of missing value in target attribute (named Clustering-based Missing value Imputation, denoted as CMI). In our approach, we fill up the missing values with plausible values that are generated by using a kernel-based method. Specifically, we first divide the dataset (including instances with missing values) into clusters. Then each instance with missing-values is assigned to a cluster most similar to it. Finally, missing values of an instance A are patched up with the plausible values generated from A's cluster.

The rest of the paper is organized as follows. In section 2, we give related work on missing values imputation. Section 3 presents our method in detail. Extensive experiments are given in Section 4. Conclusions and future work are presented in Section 5.

2   Related Work

In recent years, many researchers focused on the topic of imputing missing values. Chen and Chen [9] presented an estimating null value method, where a fuzzy similarity matrix is used to represent fuzzy relations, and the method is used to deal with one missing value in an attribute. Chen and Huang [10] constructed a genetic algorithm to impute in relational database systems. The machine learning methods also include auto associative neural network, decision tree imputation, and so on. All of these are pre-replacing methods. Embedded methods include case-wise deletion, lazy decision tree, dynamic path generation and some popular methods such as C4.5 and CART. But, these methods are not a completely satisfactory way to handle missing value problems. First, these methods only are designed to deal with the discrete values and the continuous ones are discretized before imputing the missing value, which may lose the true characteristic during the converting process from the continuous value to discretized one. Secondly, these methods usually studied the problem of missing covariates (conditional attributes).

Among missing value imputation methods that we consider in this work, there are also many existing statistical methods. Statistics-based methods include linear regression, replacement under same standard deviation, and mean-mode method. But these methods are not completely satisfactory ways to handle missing value problems. Magnani [11] has reviewed the main missing data techniques (MDTs), and revealed that statistical methods have been mainly developed to manage survey data and proved to be very effective in many situations. However, the main problem of these techniques is the need of strong model assumptions. Other missing data imputation methods include a new family of reconstruction problems for multiple images from minimal data [12], a method for handling inapplicable and unknown missing data [13], different substitution methods for replacement of missing data values [14], robust Bayesian estimator [15], and nonparametric kernel classification rules derived from incomplete (missing) data [16]. Same as the methods in machine learning, the statistical methods, which handle continuous missing values with missing in class label are very efficient, are not good at handling discrete value with missing in conditional attributes.

3   Our Algorithm

3.1   Clustering Process Strategy

The process of grouping a set of physical or abstract objects into classes of similar objects is called clustering. In this paper, we use a clustering technique, such as, K-Means [17] to group the instances of the whole dataset (denoted as S). We separate the whole database S into clusters each of which contains similar instances. When S

has more than one discrete attribute, we then use the simple matching method to compute the similarities of these discrete attributes and use the Euclidean distance to process the continuous attributes. Then the distance between instance and cluster center is a mixed one, which is a combination distances of the discrete and continuous attributes based on [6].

Our motivation in this paper is based on the assumption [6] that the instance with missing values is more likely to have the similar target attribute value as the instance that is closest to it based on the distance's principle, such as, the Euclidean distance. So we adopt the clustering method on the whole dataset in order to separate the instances into clusters based on the differences of their distances. Then the non-parametric method is utilized to deal with missing values for each cluster. Note that K=1 is a special case of K-Means method, it is the situation without clustering, that is to say, it is only a simple kernel-based imputation method while the number of clusters is 1 in our CMI algorithm. Our goal in this paper is to show the effectiveness of our method than the kernel function without clustering. Given K=1 and K>1, we can compare the performance of this non-parametric method with and without clustering the dataset. We adopt the well-known K-Means as clustering algorithm mainly for its simplicity and efficiency. As an alternative, one can choose a more powerful clustering technique for this task, for example, the G-means algorithm [19] that can determine the parameter K automatically for the clustering task.

3.2   Kernel Function Imputation Strategy

Kernel function imputation is an effective method to deal with missing values, for its computationally efficient, robust and stable [20]. In the statistical area, kernel function completion is also known as kernel nonparametric regression imputation. For instance, Zhang [20] uses the kernel method to impute missing values. In this paper, a kernel function nonparametric random imputation is proposed to make inference for the mean, variance and the distribution function (DF) of the data.

Let X be an $n \times d$-dimensional vector and let Y be a variable influenced by X, we denote X, Y as factor attributes (*FA*) (or conditional attributes) and target attribute (*TA*) respectively. We assume that X has no missing values, while only Y has. To simplify the discussion, the dataset is denoted as $(X_{i1}, X_{i2}, ..., X_{id}, Y_i, \delta_i), i = 1, ..., n$, where δ_i is an indictor function, i.e., $\delta_i = 0$ if Y_i is missing and $\delta_i = 1$ if Y_i is not missing. In a real world database, we suppose that X and Y satisfy:

$$Y_i = m(X_{i1}, X_{i2}, ..., X_{id}) + \varepsilon_i, i = 1, ..., n. \tag{1}$$

Where $m(X_{i1}, X_{i2}, ..., X_{id})$ is an unknown function, ε_i is a random error with mean 0 and variance σ^2. In other words, we assume that Y has relation with X, but we have not any idea about it. In the case of the unknown function $m(.)$ is a linear function, Wang and Rao [21, 22] show that the deterministic imputation method performance well in making inference for the mean of Y, Zhang [20] shows that one must use random imputation method in make inference for distribution functions of Y when the unknown function $m(.)$ is a arbitrary function because in many complex practical situations, the unknown function $m(.)$ is not a linear function.

In Eq.1, suppose Y_i is missing, and the value of $m(X_i)=m(X_{i1}, X_{i2},\ldots, X_{id})$ is computed by using kernel methods as follows:

$$\hat{m}(X_{i1}, X_{i2},\ldots, X_{id}) = \frac{\sum_{j=1}^{n}\delta_j Y_j \prod_{s=1}^{d} K(\dfrac{X_{is}-X_{js}}{h})}{\sum_{j=1}^{n}\delta_j \prod_{s=1}^{d} K(\dfrac{X_{is}-X_{js}}{h})+n^{-2}}, i=1,\ldots,n, \qquad (2)$$

Where $\hat{m}(X)$ is the kernel estimate of the unknown function $m(X)$ and n^{-2} is introduced

in order to avoid the case that $\sum_{j=1}^{n}\delta_j \prod_{s=1}^{d} K(\dfrac{X_{is}-X_{js}}{h})$ vanishes, and h refers

to bandwidth with $h=Cn^{-1/5}$ (we will discuss the choosing of h later in this paper). The method of using $\hat{m}(X_i)$ as imputed value of Y_i is called *kernel imputation*.

In Eq.2, $K(.)$ is a kernel function. There are many commonly used forms of kernel functions, such as the Gaussian kernel:

$$K(x) = \frac{1}{\sqrt{2\pi}}\exp(-x^2/2)$$

and the uniform kernel is presented as follows:

$$K(x) = \begin{cases} 1/2, & |x| \le 1, \\ 0, & |x| > 1. \end{cases}$$

There are not any differences for selecting the kinds of the kernel function if the optimal bandwidth can be received during the process of learning. In this paper, we adopt the widely used Gaussian kernel function.

3.3   The Strategy for Evaluating Unknown Parameters of Imputed Data

We are interesting in make inferences for the parameters of the target attribute Y such as $\mu = E(Y)$, $\sigma^2 = D(Y)$ and $\theta = F(y_0)$, i.e. the mean, the variance and the distribution function of Y, where y_0 is a fixed point, $y_0 \in R$. Based on the complete data after imputation, above parameters can be estimated as follows.

The mean of Y is given by:

$$\hat{Y} = \frac{1}{n}\sum_{i=1}^{n}\{\delta_i Y_i + (1-\delta_i) Y_i^*\} \qquad (3)$$

Where $Y_i^* = \hat{m}(X_i)$ if Y is completed by the kernel deterministic imputation method.

The variance of Y is given by:

$$\sigma^2 = \frac{1}{n}\sum_{i=1}^{n}[(\delta_i Y_i + (1-\delta_i Y_i)Y_i^*) - \hat{Y}]^2 \qquad (4)$$

In real applications, it is very difficult to work out the exact form of the distribution function of Y. So, we use the empirical form of the distribution function of Y replacing the values of the distribution function of Y:

$$\hat{F}(y_0) = \frac{1}{n} \Sigma_{i=1}^{n} I(\delta_i Y_i + (1 - \delta_i)Y_i^* \le y_0) \tag{5}$$

where $I(x)$ is the indicator function, Y_i^* is the imputed data processed by the kernel imputation methods.

3.4 Clustering-Based Missing Value Imputation Algorithm

This section presents our CMI method for missing data completion. By using the clustering techniques on the factor attributes (i.e., X), we divide the whole dataset into clusters. After clustering, we then utilize the kernel method to fill the missing-valued instance for each cluster. Note that in this paper, the kernel method is utilized to deal with the situation that Y is continuous. As for the situation of Y is discrete, we can use the nearest neighbor method (NN) to complete the missing values. Based on the above discussions, the CMI algorithm is presented as follows.

```
Procedure: CMI
Input: Missing-valued dataset S, k;
Output: Complete dataset S';
1.  (C₁,C₂,…Cₖ)←k-means(S, k);
2.  FOR each cluster Cᵢ
3.    FOR each missing-valued instance Iₖ in cluster Cᵢ
4.      use Eq. (2) to comput m̂(Xᵢ), R;
5.    FOR each missing-valued instance Iₖ in cluster Cᵢ
6.      use m̂(Xᵢ) to fill missing-value in Iₖ;
7.    S'← ∪ᵏᵢ₌₁ Cᵢ ;
```

3.5 The Choosing of c and Complexity Analysis

Kernel methods can be decomposed into two parts: one for the calculation of the kernel and another for bandwidth choice. Silverman [23] stated that one important factor in reducing the computer time is the choice of a kernel that can be calculated very quickly. Having chosen a kernel that is efficient to compute, one must then choose the bandwidth. Silverman [23] turns out that the choice of bandwidth is much more important than the choice o f kernel function. Small value of bandwidth h makes the estimate look 'wiggly' and shows spurious features, whereas too big values of h will lead to an estimate that is too smooth, in the sense, that it is too biased and may not reveal structural features. There is no generally accepted method for choosing the bandwidths. Methods currently available include 'subjective choice' and automatic methods such as the "plug-in", 'cross-validation' (CV), and 'penalizing function' approaches. In this paper we use the method of cross-validation to minimize the

approximate mean integrated square error (AMISE) of $\hat{m}(x_i)$. For a given sample of data, the CV function is defined as:

$$CV = \sum_{i=1}^{n}(y_i - \hat{m}(x_{-i},c))^2 \qquad (6)$$

where $\hat{m}(x_{-i},c)$ denotes the leave-one-out estimator evaluated for a particular value of c.

That is, the value of the missing attribute of instance i is predicted by all of the instances except instance i itself in the same class. Thus, for every missing value prediction, nearly all of the instances are selected as compared instances.

The time complexity of the kernel method is $O(n^2)$, where n is the number of instances of the dataset. After clustering, assume that the dataset is divided into k clusters, where $n_i (i = 1,2,...,k)$ is the size of cluster i. Because our CMI algorithm performs the kernel method independently on each cluster for missing value filling, so the complexity of our clustering-based kernel imputation method is $O(n_j^2)$, where n_j is the biggest number, i.e., cluster j is the largest one of all the clusters. Generally speaking, n_j is smaller than n when $k>1$, so we have $O(n_j^2) < O(n^2)$. That is, the time complexity of our method is better than the method in [20] without clustering.

4   Experimental Studies

In order to evaluate the effectiveness of our approach, we have conducted extensive experiments on datasets from the UCI machine learning repository [18]. We evaluate our algorithm on the dataset *abalone*, which contains 8 continuous attributes, one class attribute and 4177 instances in total. The other dataset is *housing* dataset, which contains 13 continuous attributes (including "class" attribute "MEDV"), one binary-valued attribute and 506 instances in total. For ease of comparison, we use random missing mechanism to generate missing values with missing rates at 5%, 20% and 40%. In the previous discussions of our strategy for handling missing values, we know that the situation of K=1 (i.e., only one cluster) is the special case, which is equal to the situation of processing the whole dataset without clustering and also similar to the kernel-based imputation method without clustering in [20].

In this paper we only report results on the mean and distribution function of *Y*. We use the AE (*average error*) to measure performance in making inference on the former two parameters:

$$AE = \frac{1}{k}\sum_{i=1}^{k}(|\hat{V_i} - V_i|/V_i) \qquad (7)$$

Where $\hat{V_i}$ is the estimated parameter (variance or empirical distribution function) value, computed from the imputed target attribute, and V_i is the parameter value of the original target attribute and k is the number of clusters.

In this paper, for the *abalone* dataset, the last attribute *Rings* is set to target attribute, others are set to factor attributes. The experimental results on *abalone* are presented in Figure 1, from (1) to (6). For the *housing* dataset, the attribute *MEDV* is set to target attribute, the results are presented in Figure 2, from (7) to (12). In these figures, 'Mean substitution' means the method of imputing missing values with mean, our method is regard as 'CMI'. In particular, it is the method in [20] while k=1 in our method, i.e., it is the method for missing values imputation without cluster.
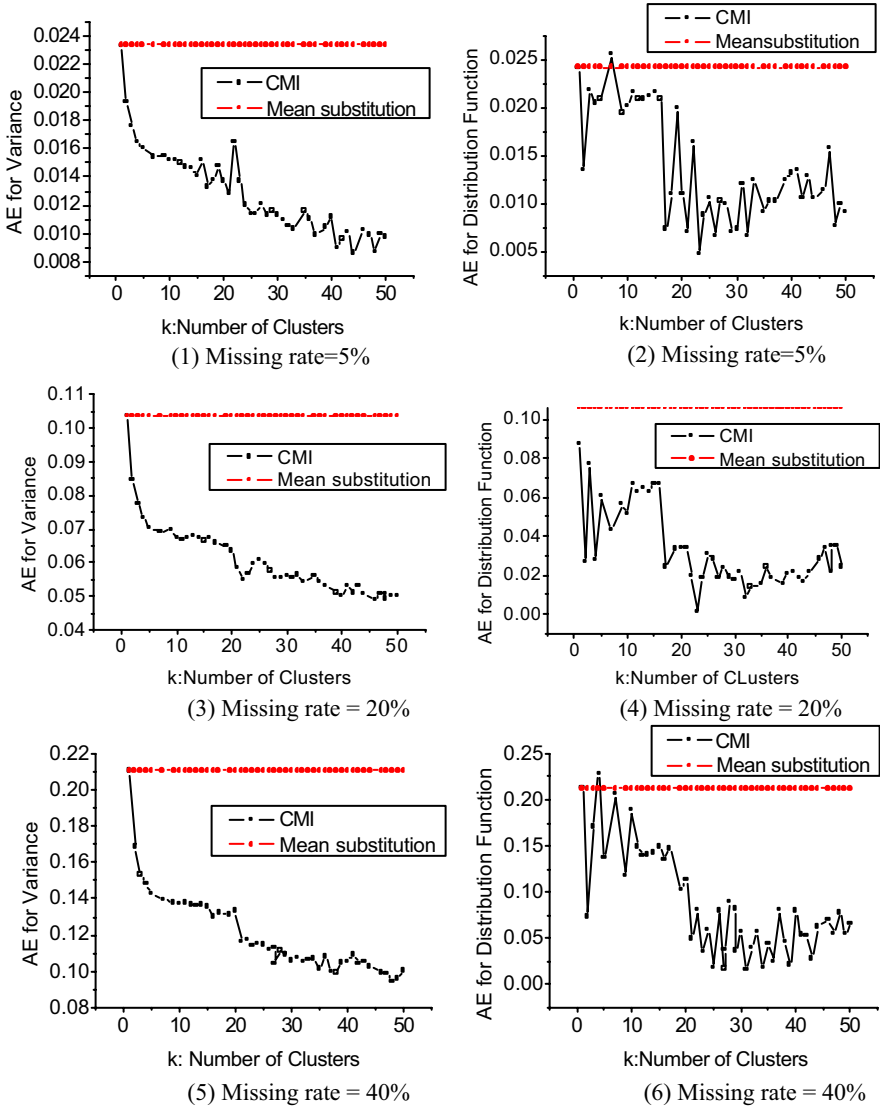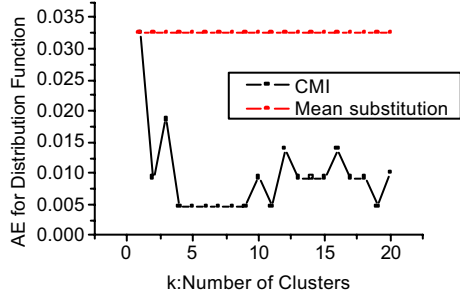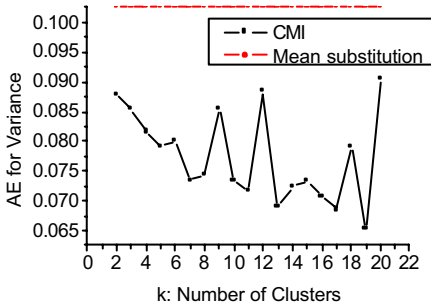

Fig. 1. CMI vs Mean substitution under different missing rates on dataset *abalone* for variance and distribution function
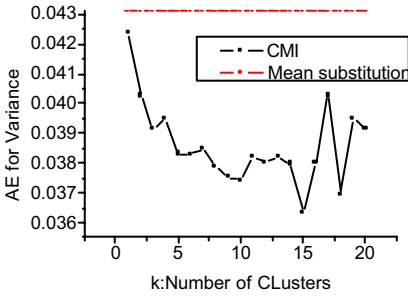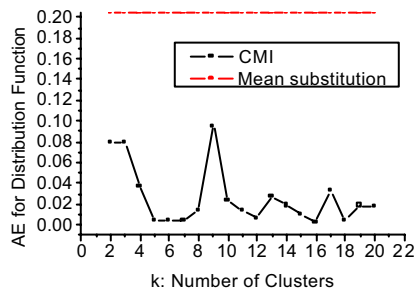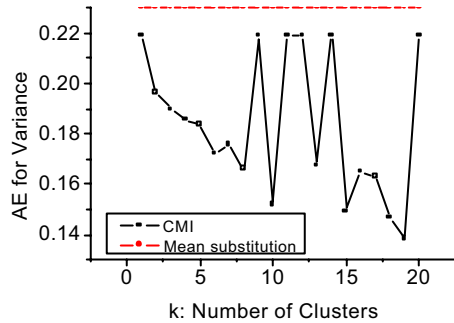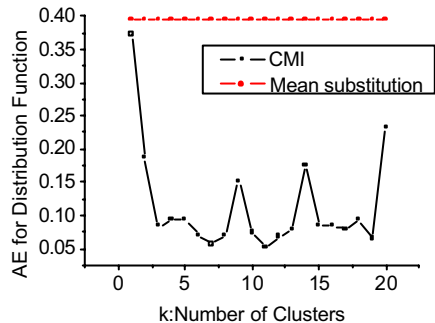
Fig. 2. CMI vs Mean substitution under different missing rates on dataset *housing* for variance and distribution function

From above figures, the method 'Mean substitution' is worst, the kernel method without clustering (k=1 in our CMI algorithm) outperform the former, and we can see that the clustering-based kernel method performs better most of the time than the kernel method that without clustering (i.e., the situation of k=1 when using k-means) in terms of variance and distribution function. All of results of our method are better than the results of the method 'Mean substitution'. What's more, with increase of

cluster number k, the average errors (AE) for variance and distribution function are decreasing. That implies that it is reasonable for us to impute missing values with cluster-based kernel methods. Yet the value of AE will increase when the number of clusters is big enough, this trend will be observed in the above figures. That is to say, the more clusters the worse performance the results of imputation are. That is because there will be less instances for imputing missing values in one cluster while the number of clusters become bigger. In our experiments, for the *Abalone* dataset (in figure 1, (1) to (6)), the best K, that is the number of clusters for K-means algorithm, ranges from 25 to 35; while for the *Housing* dataset (in figure 2, (7) to (12)), the best K ranges from 4 to 7. Note that for the large dataset, such as, the dataset *abalone* in Fig.1, the AE increases gradually while in small dataset (for instance, the dataset *Housing* in Fig.2) it increases rapidly. Because the number of instances in each cluster will change slightly when the dataset is large and there are more observed information for imputing missing values in one cluster. That makes the values of AE for variance and distribution function relatively stable compared with the previous imputation results.

These results are consistent with the results obtained by using the G-means algorithm in [19]. This means that user can use the G-means algorithm to work out the number of clusters, i.e. K, for the dataset at first, and then utilizes our CMI algorithm based on the K, in order to deal with the missing value problems on each of the cluster. As a consequence, user will be easily to choose an appropriate K for clustering in advance, without degrading the system performances for missing value imputation.

5   Conclusions and Future Work

In this paper, we propose a clustering-based non-parametric kernel-based imputation method, called CMI, for dealing with missing values, which is presented in target attribute in data preprocessing. Extensive experimental results have demonstrated the effectiveness of CMI method in making inference for variance and the distribution function after clustering. In practice, datasets usually present missing values in conditional attributes and class attributes, which makes the problem of missing value imputation more sophisticated. In our future work, we will deal with this problem.

References

1. Zhang, S.C., et al.: Information Enhancement for Data Mining. IEEE Intelligent Systems, 2004 19(2), 12–13 (2004)
2. Zhang, S.C., et al.: Missing is useful: Missing values in cost-sensitive decision trees. IEEE Transactions on Knowledge and Data Engineering 17(12), 1689–1693 (2005)
3. Qin, Y.S., et al.: Semi-parametric Optimization for Missing Data Imputation. Applied Intelligence 27(1), 79–88 (2007)
4. Zhang, C.Q., et al.: An Imputation Method for Missing Values. In: Zhou, Z.-H., Li, H., Yang, Q. (eds.) PAKDD 2007. LNCS (LNAI), vol. 4426, pp. 1080–1087. Springer, Heidelberg (2007)

5. Quinlan, J.R.: C4.5: Programs for Machine Learning. Morgan Kaufmann, San Francisco (1993)
6. Han, J., Kamber, M.: Data Mining: Concepts and Techniques, 2nd edn. Morgan Kaufmann, San Francisco (2006)
7. Chen, J., Shao, J.: Jackknife variance estimation for nearest-neighbor imputation. J. Amer. Statist. Assoc. 96, 260–269 (2001)
8. Lall, U., Sharma, A.: A nearest-neighbor bootstrap for resampling hydrologic time series. Water Resource. Res. 32, 679–693 (1996)
9. Chen, S.M., Chen, H.H.: Estimating null values in the distributed relational databases environments. Cybernetics and Systems: An International Journal 31, 851–871 (2000)
10. Chen, S.M., Huang, C.M.: Generating weighted fuzzy rules from relational database systems for estimating null values using genetic algorithms. IEEE Transactions on Fuzzy Systems 11, 495–506 (2003)
11. Magnani, M.: Techniques for dealing with missing data in knowledge discovery tasks (2004) (Version of June 2004), http://magnanim.web.cs.unibo.it/data/pdf/missingdata.pdf
12. Kahl, F., et al.: Minimal Projective Reconstruction Including Missing Data. IEEE Trans. Pattern Anal. Mach. Intell. 23(4), 418–424 (2001)
13. Gessert, G.: Handling Missing Data by Using Stored Truth Values. SIGMOD Record, 2001 20(3), 30–42 (1991)
14. Pesonen, E., et al.: Treatment of missing data values in a neural network based decision support system for acute abdominal pain. Artificial Intelligence in Medicine 13(3), 139–146 (1998)
15. Ramoni, M., Sebastiani, P.: Robust Learning with Missing Data. Machine Learning 45(2), 147–170 (2001)
16. Pawlak, M.: Kernel classification rules from missing data. IEEE Transactions on Information Theory 39(3), 979–988 (1993)
17. Forgy, E.: Cluster analysis of multivariate data: Efficiency vs. interpretability of classifications. Biometrics 21, 768 (1965)
18. Blake, C.L., Merz, C.J.: UCI Repository of machine learning databases (1998)
19. Hamerly, H., Elkan, C.: Learning the k in k-means. In: Proc. of the 17th intl. Conf. of Neural Information Processing System (2003)
20. Zhang, S.C., et al.: Optimized Parameters for Missing Data Imputation. In: Yang, Q., Webb, G. (eds.) PRICAI 2006. LNCS (LNAI), vol. 4099, pp. 1010–1016. Springer, Heidelberg (2006)
21. Wang, Q., Rao, J.: Empirical likelihood-based inference in linear models with missing data. Scand. J. Statist. 29, 563–576 (2002a)
22. Wang, Q., Rao, J.N.K.: Empirical likelihood-based inference under imputation for missing response data. Ann. Statist. 30, 896–924 (2002b)
23. Silverman, B.: Density Estimation for Statistics and Data Analysis. Chapman and Hall, New York (1986)
24. Friedman, J., et al.: Lazy Decision Trees. In: Proceedings of the 13th National Conference on Artificial Intelligence, pp. 717–724 (1996)
25. John, S., Cristianini, N.: Kernel Methods for Pattern Analysis. Cambridge (2004)
26. Lakshminarayan, K., et al.: Imputation of Missing Data Using Machine Learning Techniques. KDD 1996, 140–145 (1996)