Data & Knowledge Engineering 84 (2013) 47–58

Contents lists available at SciVerse ScienceDirect

Data & Knowledge Engineering

journal homepage: www.elsevier.com/locate/datak

An experimental study on the use of nearest neighbor-based imputation algorithms for classification tasks

Jonathan de Andrade Silva, Eduardo Raul Hruschka *

Computer Science Department, The University of São Paulo (USP) at São Carlos, Brazil

ARTICLE INFO

Article history:
Received 27 July 2011
Received in revised form 18 December 2012
Accepted 19 December 2012
Available online 4 January 2013

Keywords:
Data mining
Classification
Imputation
Missing values

ABSTRACT

The substitution of missing values, also called imputation, is an important data preparation task for data mining applications. Imputation algorithms have been traditionally compared in terms of the similarity between imputed and original values. However, this traditional approach, sometimes referred to as prediction ability, does not allow inferring the influence of imputed values in the ultimate modeling tasks (e.g., in classification). Based on an extensive experimental work, we study the influence of five nearest-neighbor based imputation algorithms (KNNImpute, SKNN, IKNNImpute, KMI and EACImpute) and two simple algorithms widely used in practice (Mean Imputation and Majority Method) on classification problems. In order to experimentally assess these algorithms, simulations of missing values were performed on six datasets by means of two missingness mechanisms: Missing Completely at Random (MCAR) and Missing at Random (MAR). The latter allows the probabilities of missingness to depend on observed data but not on missing data, whereas the former occurs when the distribution of missingness does not depend on the observed data either. The quality of the imputed values is assessed by two measures: prediction ability and classification bias. Experimental results show that IKNNImpute outperforms the other algorithms in the MCAR mechanism. KNNImpute, SKNN and EACImpute, by their turn, provided the best results in the MAR mechanism. Finally, our experiments also show that best prediction results (in terms of mean squared errors) do not necessarily yield to less classification bias.

1. Introduction

In real-world applications, datasets often contain missing values due to various reasons, such as malfunctioning of measurement equipment and non-response in surveys. Such missing data may be problematic in data mining applications. Therefore, several approaches have been proposed to deal with them [22,26]. A simple approach usually applied to handle missing values includes ignoring whole instances and/or attributes containing missing values. However, these instances and/or attributes may carry important information in the values that are present in the dataset, which can be useful to modeling tools. Although some machine learning algorithms can be tolerant to missing values, a significant number of algorithms require complete datasets. For these methods, approaches aimed at filling in missing values are particularly relevant.

The missing values problem is often circumvented via imputation. Under this perspective, an approach frequently used in practice involves replacing missing values by the mean of known values (for a quantitative attribute) or analogously by the mode (for a qualitative attribute). However, this approach considerably underestimates the population variance and tends to not preserve between-attribute relationships. Such relationships are usually explored by data mining methods. Thus, imputation algorithms should preserve them as far as possible [22]. In this sense, imputation algorithms should carefully fill in missing values,

* Corresponding author at: Department of Computer Science, The University of São Paulo (USP) at São Carlos, São Paulo, Brazil. Tel.: +55 16 3373 9700.
E-mail addresses: jandrade@icmc.usp.br (J.A. Silva), erh@icmc.usp.br (E.R. Hruschka).

0169-023X/$ – see front matter © 2012 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.datak.2012.12.006

trying to avoid the insertion of bias in the dataset. If imputation is performed in a suitable way, higher quality data becomes available, and the data mining outcomes can be improved [4].

In the last decade, several algorithms have been used for imputation [29,18,6,15,5,23]. In general, these algorithms have shown to be very useful in bioinformatics [29,6], remote sensing [2], and privacy-preserving applications [15,5]. Typically, imputation algorithms are evaluated only in terms of their prediction abilities, i.e., by computing similarities between imputed values and real, *a priori* known values. Although this approach is valid and widely adopted, another important issue that should also be addressed is about the effect of imputation on the ultimate modeling task — e.g., as discussed in [13]. In reality, the substitution process should generate values that least distorts the original characteristics of the dataset for the ultimate modeling task. In this paper, we focus on classification problems as the ultimate data mining task.

Only a few studies have addressed the influence of imputation on classification tasks. In [3,1,25,9] the authors have investigated the influence of imputation on the classification accuracy. The main idea behind their methodologies is to assess the imputation quality by means of the improvement of the classification accuracy. This approach may suggest that imputation is aimed at artificially improving the classifiers' performance. Indeed, such an improvement can be obtained by the insertion of artificial patterns into the dataset. Hruschka et al. [13] proposed a methodology to estimate the bias inserted by imputation in classification tasks. This methodology is used in our study. In brief, it assumes that both classification deterioration and improvement from datasets with imputed values are not desirable, as both imply that artificial patterns could have been incorporated into the dataset.

The main contribution of this paper is providing a thorough experimental study about the use of NN-based imputation algorithms in classification tasks. Preliminary results of this paper were reported in [7], in which we have considered only the MCAR assumption, six imputation algorithms, and five datasets. In our current work, we compare seven imputation algorithms (Mean Imputation, Majority Method [20], KNNImpute [29], SKNN [18], IKNNImpute [6], KMI [14] and EACImpute [7]) on six datasets (Iris, Glass Identification, Yeast, Pen-Digits, Segmentation, and a Synthetic dataset), with four different amounts of missing values (10%, 30%, 50% and 70%). Following the well-known Rubin's typology [24], we simulated missing values according to the distribution of missingness known as “missing completely at random” (MCAR) and “missing at random” (MAR). As discussed in great detail in [24,27], the latter allows the probabilities of missingness to depend on observed data but not on missing data, whereas the former occurs when the distribution of missingness does not depend on the observed data either. Most experiments reported in the literature consider only the MCAR assumption [26,29,3,1,25,9]. Also, we study the degree of correlation between prediction and classification results. To the best of our knowledge, previous works have not related such different aspects of the problem of assessing imputation algorithms in classification tasks. More explicitly, our study is aimed at addressing the following research questions:

- Are the performances of the algorithms under study significantly different for particular combinations of missingness mechanisms and missing rates?
- What are the correlations between prediction (e.g., mean squared error) and classification accuracy results?

The remainder of this paper is organized as follows. Section 2 reviews a methodology [13] to estimate the bias inserted by imputation algorithms in the context of classification problems. In Section 3, we briefly review the imputation algorithms that are employed in our study. The experimental results are reported in Section 4. Finally, concluding remarks are presented in Section 5.

2. Bias estimation in classification tasks

For convenience, consider that each instance i of a given dataset is described by both a vector of m attribute values $\mathbf{x}^i = [x_1^i, x_2^i, \ldots, x_m^i]$ and its corresponding class c_i, which can take any value from a set of values $C = \{c_1, c_2, \ldots, c_c\}$. A dataset can be represented by a matrix $\mathbf{X}_D$ formed by a set of vectors $\mathbf{x}^i (i = 1, \ldots, N)$, each one with an associated class $c_j \in C$, $j = 1, \ldots, c$. This matrix is formed by the values of the attributes $a_l (l = 1, \ldots, m)$ for each instance i and its respective class. Thus, x_l^i is the value of the l-th attribute of the i-th instance in $\mathbf{X}_D$. In general, $\mathbf{X}_D$ is formed by both complete instances (without any missing value) and by instances with at least one missing value. Let $\mathbf{X}_C$ be the subset of instances of $\mathbf{X}_D$ that do not have any missing value, and $\mathbf{X}_M$ be the subset of instances of $\mathbf{X}_D$ with at least one missing value, i.e., $\mathbf{X}_D = \mathbf{X}_C \cup \mathbf{X}_M$. We assume that the *class value* in $\mathbf{X}_M$ is known for every instance. In this context, imputation algorithms fill in the missing values of $\mathbf{X}_M$, originating a filled matrix $\mathbf{X}_F$.

Following [13], in an ideal situation, the imputation algorithm fills in the missing values, originating filled values, without inserting any bias in the dataset. In a more realistic view, imputation algorithms are aimed at decreasing the amount of inserted bias to acceptable levels, in such a way that a dataset $\mathbf{X'}_D = \mathbf{X}_C \cup \mathbf{X}_F$, probably containing more information than $\mathbf{X}_C$, can be used for data mining (e.g., considering issues such as attribute selection, combining multiple models, and so on). From this standpoint, it is particularly important to emphasize that we are assuming that the known values in $\mathbf{X}_M$ may contain important information for the modeling process. This information would be (partially) lost if the instances and/or attributes with missing values were ignored.

Two general approaches have been used in the literature to evaluate the bias inserted by imputations. We shall refer to them as *prediction* and *modeling* approaches. In a prediction approach, missing values are simulated, e.g., some known values are removed and then imputed. For instance, some known values from $\mathbf{X}_C$ could be artificially eliminated, simulating missing entries. In this way, it is possible to evaluate how similar the imputed values are to the real, *a priori* known values — e.g., using the popular mean squared error. The underlying assumption behind this approach is that the more similar the imputed value is to the real value, the better the imputation algorithm is. Although the prediction approach is valid and widely adopted, the prediction results

are not the most important issue to be analyzed as discussed, for instance, in [13]. In brief, the prediction approach does not allow estimating the inserted bias from a modeling perspective. More precisely, the substitution process must generate values that least distorts the original characteristics of $\mathbf{X}_D$, which can be assumed to be the between-attribute relationships, for the modeling process. These relationships are often explored by classification algorithms. For the sake of argument, let us consider the pedagogical example depicted in Fig 1. This example is inspired from the widely known Iris dataset, which contains instances formed by 4 attributes (SL, SW, PL, and PW) and the class label. Consider that the instance whose ID is *151* has a missing value for attribute PW. This instance is identical to instance *44*, except for the missing value. In other words, instance *151* could be viewed as a result of a procedure for missing value simulation widely used in the literature. In this context, an imputation algorithm should estimate a value as close as possible to 0.6, which is the known value artificially excluded from the instance *44*. Let us now assume that two imputation algorithms (A and B) are available to substitute such a missing value. Also, consider that algorithm A substitutes the missing value by 0.2, whereas algorithm B substitutes it by 0.601. Clearly algorithm B is better than algorithm A from the prediction point of view. However, consider now that the tree depicted in Fig 1 is a perfect classifier. According to this classifier, imputation algorithm B would make instance *151* to be incorrectly classified, whereas imputation algorithm A, which is not as good as B from the prediction point of view, would lead to the correct classification of the considered instance.

Several authors – e.g., see [13] for a review – have also argued that it is more important to take into account the influence of imputed values in the modeling process (e.g., preserving the relationships between attributes) than to get more accurate predictions. Indeed, although the imputed values are predictions, it is not the accuracy of these predictions that is of most importance when replacing missing values. It is more important that such predictions produce a workable estimate that least distorts the values that are actually present in the dataset. In other words, the main purpose of imputation is not to use the values themselves, but to make available to the modeling tools the information contained in the other variables' values that are present in the dataset. For all these reasons, we have focused on the inserted biases in terms of classification results, which somehow allow evaluating to what extent the relationships between attributes are being maintained after imputation. Finally, one must acknowledge that in real-world applications the imputed values cannot be compared with any value.

The bias inserted by imputation can be defined as [13] "*the magnitude of the change in the between-attribute relationships caused by patterns introduced by an imputation process*". The problem is that the relationships between attributes are hardly known a priori, before data mining is performed. Therefore, usually the inserted bias cannot be directly measured, but it can be estimated. In classification problems, the underlying assumption is that between-attribute relationships are induced by a particular classifier. Consequently, the quality of such discovered relationships can be indirectly estimated by classification measures like the Average Correct Classification Rate (ACCR). In this sense, we adopt a methodology to estimate the inserted bias detailed in [13] and addressed in the sequel.

In data mining applications, different classifiers are often assessed for a given dataset, in such a way that the best available classifier is then chosen according to some criterion of model quality (e.g., the ACCR). Our underlying assumption is that the best classifier (BC) – in relation to $\mathbf{X}_C$ and to the available classifiers – provides a suitable model for classifying instances after imputations have been performed. Thus, it is important to assess if the imputed values adjust themselves to the BC model. It is a common practice to evaluate classifier performance in a test set. The same concept can be adapted to evaluate imputations, considering $\mathbf{X}_C$ as the training set and $\mathbf{X}_F$ as the test set. Then, inserted bias can be estimated by means of the procedure in Fig. 2. According to this procedure, a positive bias is achieved when the ACCR in $\mathbf{X}_F$ (step 2) is greater than in the cross-validation process in $\mathbf{X}_C$ (step 1). In this case, the imputed values are likely to improve the classifier's ACCR in X'_D. Accordingly, a negative bias is inserted when the imputed values are likely to worsen the classifier's ACCR in X'_D. Finally, no bias is likely inserted when the classifier's accuracies in $\mathbf{X}_F$ and in the cross-validation process in $\mathbf{X}_C$ are equal. Assuming that the imputation process should not introduce artificial patterns into the data, this is the ideal situation. Indeed, these artificial patterns, not present in the known values, may be later discovered during the data mining process in X'_D. Therefore, the inclusion of such artificial patterns, which are

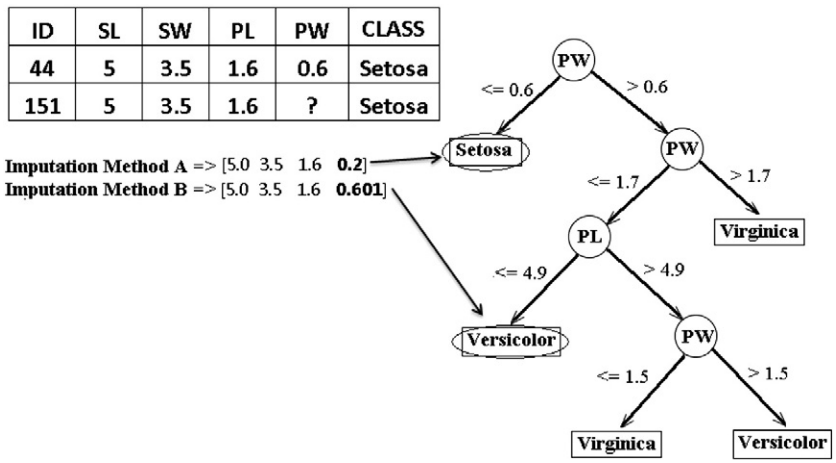

Fig. 1. Pedagogical example of bias in classification.

1)Evaluate the classifier's ACCR by cross-validation in X_C, obtaining ACCRc;
2)Evaluate the classifier's ACCR in X_F (here viewed as a test set) considering that X_C is the training set.
In other words, this step involves building the classifier in X_C and then testing it in the instances of X_F,
thus obtaining ACCRF.
3)The bias ($\widehat{b}$) inserted by the performed imputations is estimated from the difference between the results achieved in steps 2) and 1): $\widehat{b}$ = ACCRF − ACCRc.

Fig. 2. Estimating the inserted bias on classification.

simply an artifact of the imputation process, should be avoided. According to this elaboration, not only a negative bias but also a positive bias is not desirable, as both imply that artificial patterns have been likely incorporated into the dataset.

3. NN-based imputation algorithms

In this section we briefly review the NN-based imputation algorithms used in our comparative study: KNNImpute [29], SKNN [18], IKNNImpute [6], KMI [14] and EACImpute [7]. Essentially, NN-based imputation algorithms find a subset of instances that are the most similar to the instance with missing values. Then, this subset of nearest instances is used to replace the missing values [29,3,18,14]. Also, two simple algorithms (Mean Imputation and Majority Method [20]) were also used. In classification problems, imputation algorithms can be adapted to take into account the information provided by the class. In this sense, the NN-based imputation algorithms can be adapted for supervised imputation, which involves using them in the instances of each class separately, as done in this work.

3.1. KNNImpute

KNNImpute is a popular imputation algorithm introduced in [29]. This algorithm is available in systems widely used in practice, such as SAM[1] [30], PAM[2] [28] and MAANOVA[3] [16]. Briefly, KNNImpute chooses *k*-nearest instances that necessarily contain observed values on the respective attribute of the instance that contains the missing value to be imputed. Usually, the Euclidean distance is used to calculate the dissimilarity between two instances — considering only pairs of values that are not missing. The missing value is replaced by the weighted average over *k*-nearest instances, where the contribution (weight) of each nearest instance is inversely proportional to its distance to the instance that contains missing values.

3.2. SKNN

Kim et al. [18] proposed a variant of KNNImpute known as Sequential KNN (SKNN). This algorithm sequentially imputes missing values from the instance with the least amount of missing values, and also uses the imputed values for later imputations. Initially, the dataset is split into two sets that contain complete instances (complete set) and instances with missing values (incomplete set), in which the instances are ordered by their missing rate. For each instance having the smallest missing rate in the incomplete set, SKNN finds the *k*-nearest instances in the complete set. The missing values are then filled according to KNNImpute rule just described. After imputation, the complete set is updated.

3.3. IKNNImpute

Brás and Menezes [6] proposed a variant of KNNImpute, known as Iterative KNN Imputation (IKNNImpute). This algorithm is based on a procedure that initially involves replacing all missing values via Mean Imputation and iteratively refining these estimates by the NN-based imputation principle. The NN-based imputation algorithm selects the *k* nearest instances that are the most similar to the instance previously imputed. The dataset is considered completely imputed when the difference between imputed values in two consecutive iterations is less than a threshold specified in advance.

[1] www.stat.stanford.edu/~tibs/SAM.
[2] www.stat.stanford.edu/~tibs/PAM.
[3] http://research.jax.org/faculty/churchill/software/Rmaanova.

3.4. KMI

K-Means Imputation (KMI) [14] is explicitly based on the data clustering concept. This algorithm consists basically of two steps. Initially, KMI performs a clustering step on the complete set. In this step, the well-known *K*-Means algorithm is employed to obtain representative instances (centroids), which are used as entries to the imputation step. The imputation step consists in finding the closest centroid to each instance in the incomplete set. Then, the closest centroid's values are used to fill in missing values.

3.5. EACImpute

An Evolutionary Algorithm for Clustering-based Imputation – EACImpute – has been proposed in [7]. This algorithm relies on the assumption that clusters of (partially unknown) data can provide useful information for imputation purposes. The underlying idea behind EACImpute is to repeatedly perform a clustering step followed by an imputation step. The clustering step is based on evolutionary search process that evolves data partitions with variable number of clusters by eliminating, splitting, and merging clusters that are systematically refined by *K*-Means (adapted to deal with missing values). Imputations are performed by means of the nearest-neighbor principle. The missing values are filled in by taking into account all neighbor instances belonging to the same cluster, which leads to an automatic determination of the number of neighbors (cluster size).

4. Experimental evaluation

4.1. General study design

In order to compare the algorithms addressed in Section 3 – Mean Imputation, Majority Method [20], KNNImpute [29], SKNN [18], IKNNImpute [6], KMI [14] and EACImpute [7] – we employed six datasets that are widely used in machine learning and data mining studies – Iris, Glass Identification, Yeast, Pen-Digits, and Segmentation – as well as a complementary synthetic dataset. Section 4.1.1 describes their main characteristics.

For the adopted datasets, we simulated four different amounts of missing values (10%, 30%, 50% and 70%) for each missingness mechanism — MCAR and MAR. Recall from the introduction that MAR allows the probabilities of missingness to depend on observed data but not on missing data, whereas MCAR occurs when the distribution of missingness does not depend on the observed data either. Following [27], "*when missing values occur for reasons beyond our control, we must make assumptions about the processes that create them. These assumptions are usually untestable*." The authors then suggest that assumptions should be made explicit and the sensitivity of results be investigated. In this context, we report a variety of results from different MCAR and MAR assumptions. Further details on how missing values have been simulated are addressed in Section 4.1.2.

We are interested in assessing the relative performance of the algorithms under investigation according to two measures commonly used in the literature, namely: *Normalized Root Mean Squared Error* (NRMSE) and Classification Accuracy (see Section 4.1.3). In particular, we studied the degree of correlation between NRMSE and classification accuracy.

Finally, Section 4.1.4 addresses the procedure adopted for performing the statistical analysis of the obtained results, and Section 4.1.5 reports the adopted parameter settings.

4.1.1. Datasets

In our experiments, we used five datasets from the UCI Machine Learning Repository[4] (Iris, Glass Identification, Yeast, Segmentation, Pen-Digits), and a Synthetic dataset [10], whose main features are summarized in Table 1. Originally, these datasets do not contain missing values. Since we are using Euclidean distance to compute dissimilarities, only the quantitative attributes were considered in the experiments.

4.1.2. Missingness mechanisms

Simulations of missing values have been designed to remove values from numerical attributes according to different rates, viz. 10%, 30%, 50%, and 70% in a supervised way (i.e., conditioned on the class values). For each of these missing rates, 30 repetitions were performed in the following way:

- MCAR: Missing values were generated for different quantities of attributes (1, 2,..., $m/2$), where m is the number of attributes (see third column of Table 1), resulting in 240, 480, 480, 1080 and 960 datasets with missing values respectively to Iris, Glass Identification, Yeast, Segmentation and Pen-Digits. Fig. 3 illustrates this process by considering a dataset with two classes. This dataset was split into two subsets, and missing values were randomly introduced 30 times for each missing rate (with different amounts of attributes with missing values).
- MAR: In this mechanism, we used a feature selection procedure based on the so-called Wrapper approach [19], which was initially applied by using the popular C4.5 decision tree learning algorithm. This procedure was adopted to insert missing values, in a controlled manner, into attributes that are potentially more discriminative for classification purposes. To that end, we need to know the range of values for the selected attributes that would be better for discriminating the classes. These values can be

[4] A. Asuncion and D. Newman, 2007. UCI datasets available at: http://www.ics.uci.edu/mlearn/MLRepository.html.

Table 1
Summary of datasets.

Datasets	# Instances	# Attributes	# Classes
Iris	150	4	3
Glass Identification	214	9	6
Yeast	1484	8	10
Segmentation	2310	19	7
Pen-Digits	10,992	16	10
Synthetic	500	2	3

obtained, for example, from the decision tree. In particular, missing values were introduced into the most relevant attribute (the root of the decision tree) by considering specific ranges of values for the remaining attributes chosen by C4.5. The missing values simulation for this mechanism is illustrated in Fig. 4, in which a_1^* is the most relevant feature (attribute). Right below the right hand tables there is an example of condition used to insert missing values into the attribute a_1^*, where x and y are discriminatory values belonging to the a_2 and a_3 domains, respectively. For simplicity, the illustrated condition is deterministic, but we have actually used probabilistic conditions, indirectly captured by the employed missingness rates. These simulations were performed on Synthetic, Iris, and Segmentation datasets, resulting in 360 datasets with missing values. The remaining datasets were not used to simulate the MAR mechanism because, by using the adopted procedure, it was not clear how to identify the required ranges of potentially interesting values to simulate missing entries in those datasets.

4.1.3. Evaluation measures

We evaluated the performance of the studied imputation algorithms by means of two measures: prediction accuracy and classification bias. The accuracy of prediction was evaluated by calculating the error between real (known) values and the respective imputed values by using the widely known *Normalized Root Mean Squared Error* (NRMSE) [21,17,6]. The classification bias was estimated by means of the procedure detailed in Section 2. To do so, it would be desirable to employ a classifier as accurate as possible for each of the datasets in hand. Therefore, in order to estimate the classification bias inserted by the performed imputations, four classifiers that are popular in the data mining community were used: C4.5 (J4.8), *k*-Nearest-Neighbors (KNN), Multilayer Perceptron (MLP), and Naïve Bayes (NB). These classifiers make part of the WEKA System [11], which was used to perform our experiments (using its default parameters). These classifiers were previously assessed in a 10-fold cross validation process, and ranked according to their Average Correct Classification Rates (ACCR's) — see Table 2. Then, for each dataset, the best classifier is used for estimating the classification bias inserted by imputations. For example, J4.8 is used to estimate the classification bias for Pen-Digits.

4.1.4. Statistical analysis

We statistically analyzed the NRMSE and classification bias results by following the approach proposed by [8]. In brief, this approach is aimed at comparing multiple algorithms on multiple datasets, and it is based on the use of the well-known Friedman test with a corresponding post-hoc test. The Friedman test is a non-parametric statistic test equivalent to the repeated-measures ANOVA. If the null hypothesis, which states that the algorithms under study have similar performances, is rejected, then we proceed with the Nemenyi post-hoc test [12] for pair-wise comparisons between algorithms.

For each missingness configuration (missingness rate and number of attributes), 30 datasets with missing values filled in by seven imputation algorithms were evaluated by using the NRMSE and the classification bias. Our experiments include four different rates of missing values, variable amounts of attributes with missing values, and two missingness mechanisms, totalizing 3,600 different scenarios that were statistically analyzed. For compactness, we here only report a summary of the obtained results. In brief, rank values are assigned for each measure. In particular, the best performing algorithm is assigned the rank of 1, the second best the rank of 2, and so forth. Then, the Friedman test checks whether the measured average ranks are significantly different from the mean rank expected under the null-hypothesis (all algorithms perform equally well). The Nemenyi test

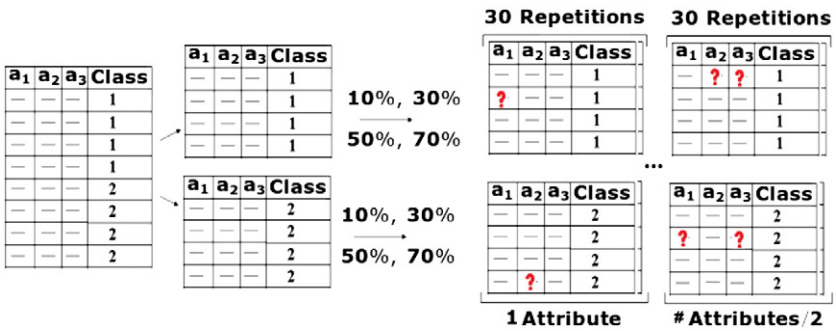

Fig. 3. Simulation process of missing values by MCAR mechanism.

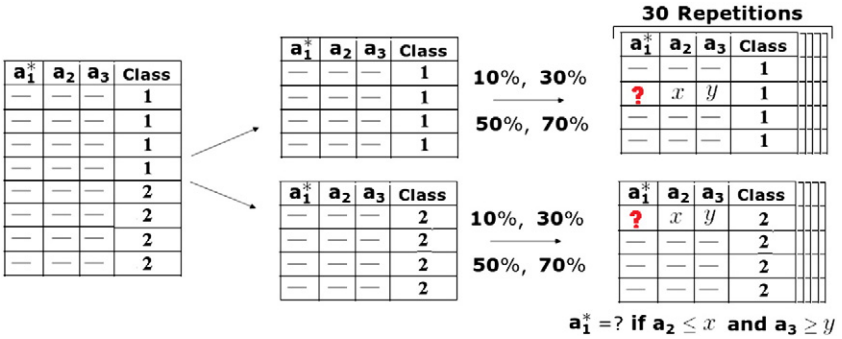

Fig. 4. Simulation process of missing values by MAR mechanism. The symbol "*" indicates the selected attribute.

indicates which pairs of algorithms are significantly different. The statistical results obtained for each algorithm, in all possible pair-wise comparisons, can be summarized by means of win/tie/loss tables. We calculated the number of times that an algorithm was statistically superior (win), inferior (loss), and similar (tie) for every pair of missing rate and dataset. Fig. 5 illustrates the procedure adopted for performing statistical analysis.

4.1.5. Parameters settings

We run KNNImpute [29], SKNN [18] and IKNNImpute [6] by setting their parameters as suggested by the original references. In particular, for KNNImpute, SKNN, and IKNNImpute, the number of neighbors was set to 10. In addition, the number of iterations was set to 2 for IKNNImpute. Two algorithms that perform clustering based imputation (KMI and EACImpute) were also evaluated. For KMI, we set the number of clusters and neighbors to 2 and 1, respectively. For EACImpute, we adopted the following parameters as defined in [7]: populations formed by 5 genotypes, generations $G = 20$ and 5 iterations for K-Means.

4.2. Results

From the study design described in Section 4.1, we are now in position to report our obtained results. Sections 4.2.1 and 4.2.2 report the achieved results for MCAR and MAR mechanisms, respectively.

4.2.1. MCAR

Fig. 6 summarizes the average results (over five datasets, different number of attributes, and 30 repetitions) obtained for both the NRMSE and classification bias. One can observe that KNNImpute, SKNN, and IKNNImpute have shown the best results in terms of the NRMSE. As expected, Mean Imputation presented the worst results. Considering the classification bias, it is worth noticing that all algorithms presented similar performances, except for the Mean Imputation, which has shown significantly worse results than its counterparts.

Let us now analyze the obtained results by taking into account the procedure described in Section 4.1.4. Fig. 7 shows the frequency that an algorithm is statistically superior or equal to the other algorithms by considering 20 possible cases (from 5 datasets and 4 missing rates). The bars in this figure show the frequency that an algorithm was considered the best for each measure (NMRSE and classification bias). One can observe that IKNNImpute provides superior NRMSE results when compared with the other imputation algorithms. Considering the classification bias, IKNNImpute still achieves better performance than the other algorithms (although with little difference to KNNImpute). We also observed that the Majority Method presented competitive results in relation to SKNN, KMI, and EACImpute.

We have also analyzed the Pearson's correlation between NMRSE and classification bias. For illustration purposes, we show in Fig. 8 the scatter plot for the Pen-Digits dataset, in which IKNNImpute has shown the best overall performance in terms of the NRMSE values. The respective Pearson's correlation coefficient is $\rho = 0.2838$, which suggests that better prediction results do not

Table 2
Average Correct Classification Rates (ACCRs) — best results in bold.

Datasets	Classifiers			
	J4.8	*k*-NN	MLP	NB
Iris	94.7%	95.2%	**96.9%**	95.5%
Glass Identification	67.6%	**70.0%**	67.3%	49.4%
Yeast	56.4%	54.3%	**58.8%**	57.9%
Segmentation	96.4%	**99.3%**	94.6%	85.8%
Pen-Digits	**96.8%**	96.1%	96.0%	80.3%

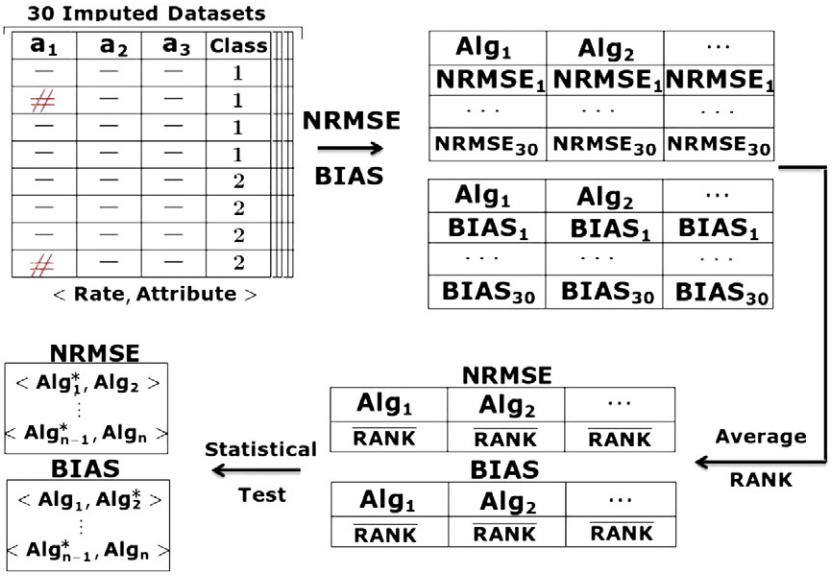

Fig. 5. Statistical analysis process. The symbols "#" and "*" represent the imputed values and the algorithm statistically superior in pair-wise comparisons, respectively.

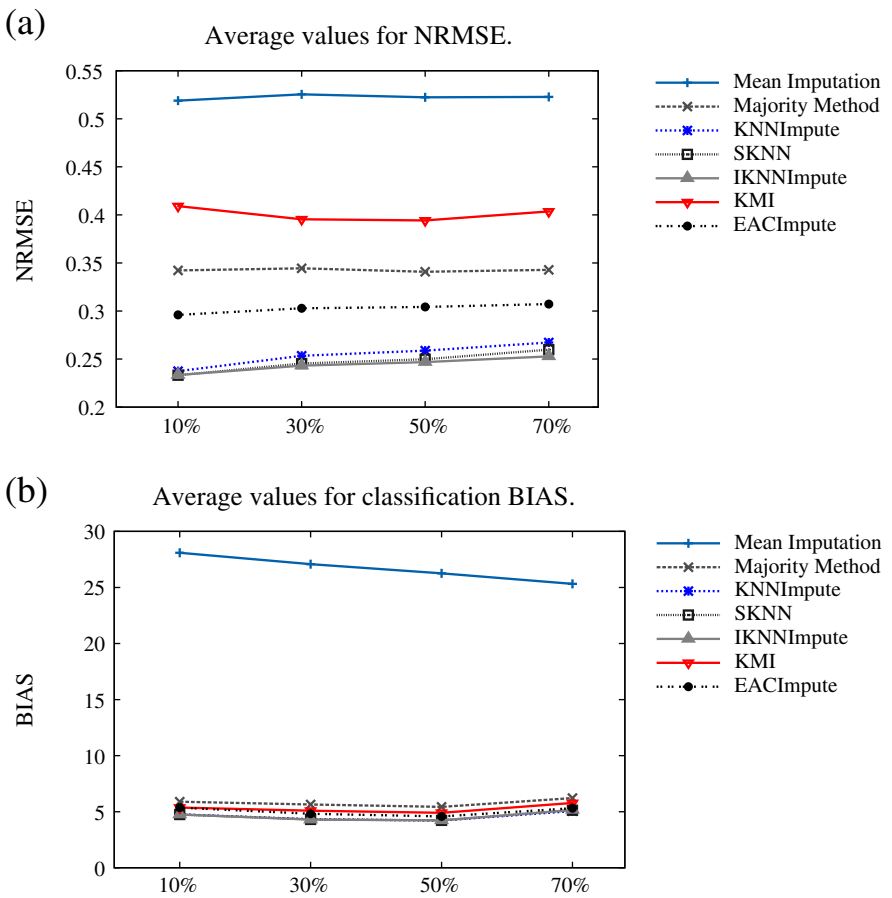

Fig. 6. Results of NRMSE and classification BIAS — MCAR.

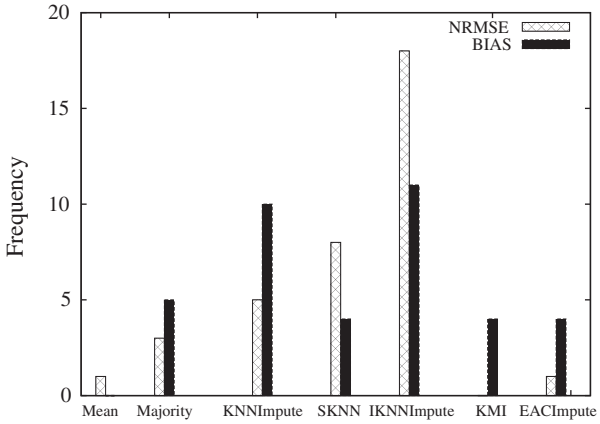

Fig. 7. Summary of statistical results for MCAR (frequency that an algorithm is statistically superior or equal to the others).

necessarily imply less classification bias. For the other datasets, ρ values range from -0.1164 to 0.2034, leading to analogous conclusions.

4.2.2. MAR

Fig. 9 summarizes the average results (over three datasets and 30 repetitions) obtained for both the NRMSE and classification bias. In brief, SKNN, KNNImpute, and EACImpute have shown the best results in terms of the NRMSE. As expected, the Majority Method presented the worst results. Considering the classification bias, it is worth noticing that all algorithms presented similar performances, except for the Majority Method, which has shown significantly worse results than the other algorithms.

Let us now analyze the obtained results by taking into account the procedure described in Section 4.1.4. In this mechanism, we did not consider Mean Imputation due to its poor performance in MCAR simulations. Following analyses similar to those performed for the MCAR mechanism, Fig. 10 shows the frequency that an algorithm is statistically superior or equal to the others. We can see that SKNN and EACImpute provided better NRMSE results when compared with the other imputation algorithms. In what concerns the classification bias, KNNImpute, SKNN, and EACImpute achieved better performance. For illustrating the correlation between NMRSE and classification bias, Fig. 11 shows the scatter plot for the Synthetic dataset, in which KMI frequently provided the lowest NRMSE values. Similar results were obtained for the other datasets, again suggesting that better prediction results do not necessarily lead to less classification bias.

5. Conclusions

This paper presented an experimental study on the use of nearest-neighbors based imputation algorithms in classification tasks. Seven imputation algorithms were used to impute missing values in six datasets with different amounts of missing values (10%, 30%, 50% and 70%) artificially introduced by MCAR and MAR mechanisms. We statistically analyzed 3600 different scenarios to assess these imputation algorithms by the so-called *Normalized Root Mean Squared Error* (NRMSE), as well as by estimating the classification bias. Experimental results in the MCAR mechanism showed that the IKNNImpute algorithm [6] achieved the best results for both NRMSE and classification bias. In the MAR mechanism, KNNImpute [29], SKNNImpute [18], and EACImpute [7]

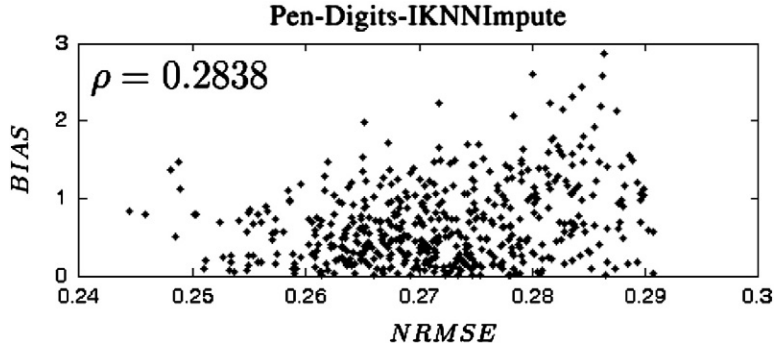

Fig. 8. Correlation between NRMSE and classification BIAS — IKNNImpute in the Pen-Digits dataset.

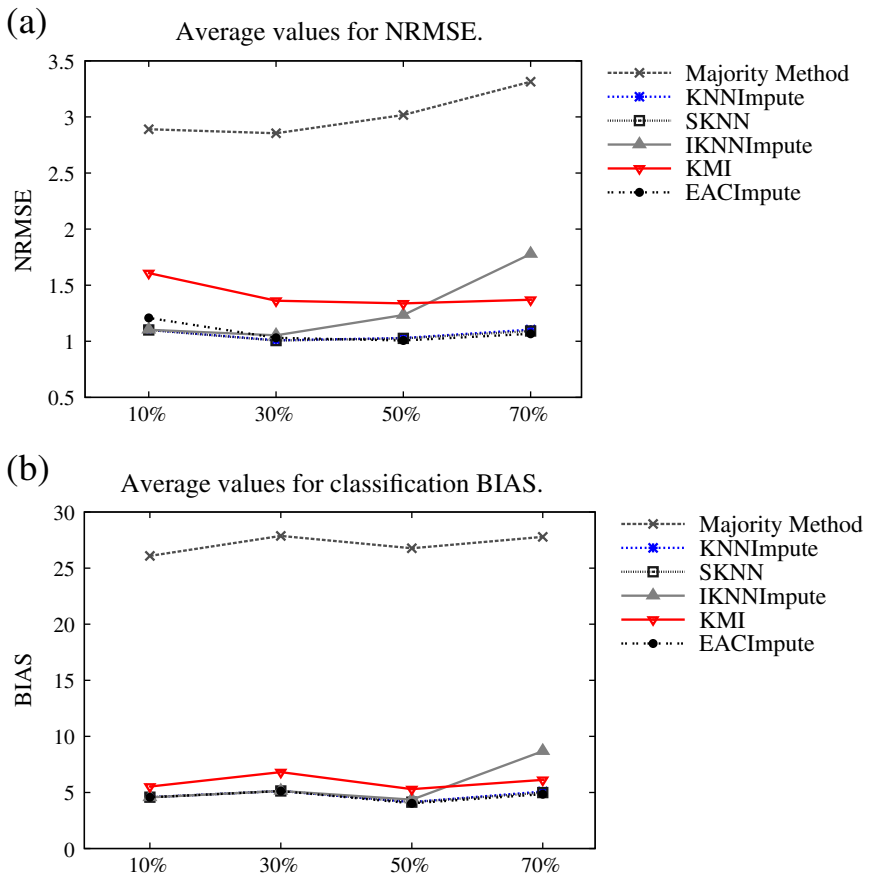

Fig. 9. Results of NRMSE and classification BIAS — MAR.

showed better results than the other algorithms. The low correlations observed between NRMSE and classification bias suggest that best prediction results (NRMSE) do not necessarily lead to best results in terms of classification bias. Promising future work includes the study of a measure to estimate the classification bias that explicitly takes into account the sample variability.

Acknowledgments

The authors would like to thank the Research Agencies CAPES, CNPq, and FAPESP for their financial support.

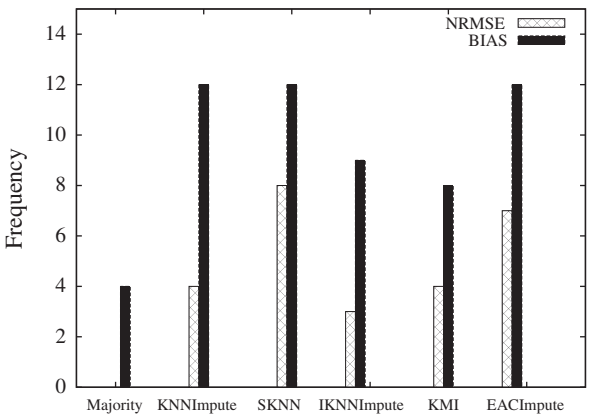

Fig. 10. Summary of statistical results for MAR (frequency that an algorithm is statistically superior or equal to the others).

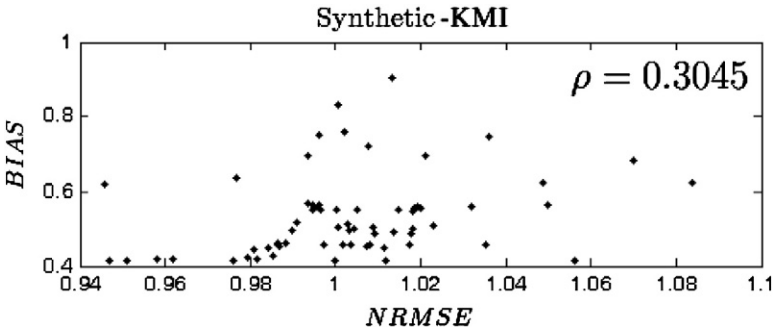

Fig. 11. Correlation between NRMSE and classification BIAS — KMI in the Synthetic dataset.

References

[1] E. Acuna, C. Rodriguez, The treatment of missing values and its effect in the classifier accuracy, Classification, Clustering and Data Mining Applications (2004) 639–648.
[2] A. Barth, J. Wallerman, G. Stahl, Spatially consistent nearest neighbor imputation of forest stand data, Remote Sensing of Environment 113 (March 2009) 546–553.
[3] G.E.A.P.A. Batista, M.C. Monard, An analysis of four missing data treatment methods for supervised learning, Applied Artificial Intelligence (2003) 519–533.
[4] R. Blake, P. Mangiameli, The effects and interactions of data quality and problem complexity on classification, Journal Data and Information Quality 2 (2011) 1–28.
[5] F. Bonchi, B. Malin, Y. Saygin, Recent advances in preserving privacy when mining data, Data & Knowledge Engineering 65 (2008) 1–4.
[6] L.P. Brás, J.C. Menezes, Improving cluster-based missing value estimation of DNA microarray data, Biomolecular Engineering 24 (Junho 2007) 273–282.
[7] J. de Andrade Silva, E.R. Hruschka, EACImpute: an evolutionary algorithm for clustering-based imputation, 5th International Conference on Intelligent Systems Design and Applications, IEEE Press, Pisa, Italy, 2009, pp. 1400–1406.
[8] J. Demšar, Statistical comparisons of classifiers over multiple data sets, Journal of Machine Learning Research 7 (January 2006) 1–30.
[9] A. Farhangfar, L. Kurgan, J. Dy, Impact of imputation of missing values on classification error for discrete data, Pattern Recognition (2008) 3692–3705.
[10] A.J.T. Garcia, E.R. Hruschka, Naive Bayes as an imputation tool for classification problems, IEEE Press, Los Alamitos, CA, USA, 2005, pp. 497–499.
[11] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, I.H. Witten, The weka data mining software: an update, SIGKDD Explorations Newsletter (2009) 10–18.
[12] M. Hollander, D.A. Wolfe, Nonparametric statistical methods, A Wiley Publication in Applied Statistics, 2nd edition, Wiley, New York, 1999.
[13] E.R. Hruschka, A.J.T. Garcia, E.R. Hruschka Jr., N.F.F. Ebecken, On the influence of imputation in classification: practical issues, Journal of Experimental and Theoretical Artificial Intelligence (2009) 43–58.
[14] E.R. Hruschka, E.R. Hruschka Junior, N.F.F. Ebecken, Towards efficient imputation by nearest-neighbors: a clustering-based approach, 6th Australian Conference on Artificial Intelligence, Spring-Verlag, 2004, pp. 513–525.
[15] G. Jagannathan, R.N. Wright, Privacy-preserving imputation of missing data, Data & Knowledge Engineering 65 (2008) 40–56.
[16] M.K. Kerr, M. Martin, G.A. Churchill, Analysis of variance for gene expression microarray data, Journal of Computational Biology (2000) 819–837.
[17] H. Kim, G.H. Golub, H. Park, Missing value estimation for DNA microarray gene expression data: local least squares imputation, Bioinformatics (2005) 187–198.
[18] K.-Y. Kim, B.-J. Kim, G.-S. Yi, Reuse of imputed data in microarray analysis increases imputation efficiency, BMC Bioinformatics 5 (2004) 160.
[19] R. Kohavi, G.H. John, Wrappers for feature subset selection, Artificial Intelligence 97 (1–2) (1997) 273–324.
[20] I. Kononenko, I. Bratko, E. Roskar, Experiments in automatic learning of medical diagnostic rules, Tech. rep., Jozef Stefan Institute, Ljubjana, Yogoslavia, 1984.
[21] S. Oba, M.-A. Sato, I. Takemasa, M. Monden, K.I. Matsubara, S. Ishii, A Bayesian missing value estimation method for gene expression profile data, Bioinformatics 16 (November 2003) 2088–2096.
[22] D. Pyle, Data Preparation for Data Mining (The Morgan Kaufmann Series in Data Management Systems), Morgan Kaufmann, March 1999.
[23] Y. Ren, G. Li, J. Zhang, W. Zhou, The efficient imputation method for neighborhood-based collaborative filtering, Proceedings of the 21st ACM international conference on Information and knowledge management, CIKM '12ACM, 2012, pp. 684–693.
[24] D.B. Rubin, Formalizing subjective notion about the effects of nonrespondents in samples surveys, Journal of the American Statistical Association (1977) 538–543.
[25] M. Saar-Tsechansky, F. Provost, Handling missing values when applying classification models, Journal of Machine Learning Research (2007) 1623–1657.
[26] J. Schafer, Analysis of Incomplete Multivariate Data, Chapman & Hall/CRC, 2000.
[27] J.L. Schafer, J.W. Graham, Missing data: our view of the state of the art, Psychological Methods 7 (2002) 147–177.
[28] R. Tibshirani, T. Hastie, D. Narasimhan, G. Chu, Diagnosis of multiple cancer types by shrunken centroids of gene expression, Proceedings of the National Academy of Sciences, 2002, pp. 6567–6572.
[29] O.G. Troyanskaya, M. Cantor, G. Sherlock, P.O. Brown, T. Hastie, R. Tibshirani, D. Botstein, R.B. Altman, Missing value estimation methods for DNA microarrays, Bioinformatics 6 (2001) 520–525.
[30] V. Tusher, R. Tibshirani, G. Chu, Significance analysis of microarrays applied to the ionizing radiation response, Proceedings of the National Academy of Sciences, 2001, pp. 5116–5121.

Jonathan de Andrade Silva received his B.Sc. degree in Computer Engineering from Catholic University Dom Bosco, Brazil, in 2007 and his M.Sc. degree from University of São Paulo, Brazil, in 2010. He is currently a Ph.D. Candidate in the Institute of Mathematics and Computer Science (ICMC) at the University of São Paulo. His research interests are in the areas of evolutionary computation, mining data stream, missing values imputation, and clustering algorithms.

Eduardo Raul Hruschka received his Ph.D. degree in Computational Systems from Federal University of Rio de Janeiro, Brazil, in 2001. He is currently associate professor of the Computer Science Department of the University of São Paulo (USP) at São Carlos. His primary research interests are in data mining. He has authored or coauthored more than 60 research publications in peer-reviewed reputed journals and conference proceedings. Dr. Hruschka is serving as associate editor of the journal Information Sciences (Elsevier). He has also been working as a reviewer for several journals, as well as a member of the Program Committee of several international conferences.