Knowl Inf Syst
DOI 10.1007/s10115-011-0424-2

REGULAR PAPER

On the choice of the best imputation methods for missing values considering three groups of classification methods

Julián Luengo · Salvador García · Francisco Herrera

Received: 24 June 2009 / Revised: 10 March 2011 / Accepted: 22 May 2011

Abstract In real-life data, information is frequently lost in data mining, caused by the presence of missing values in attributes. Several schemes have been studied to overcome the drawbacks produced by missing values in data mining tasks; one of the most well known is based on preprocessing, formerly known as imputation. In this work, we focus on a classification task with twenty-three classification methods and fourteen different imputation approaches to missing values treatment that are presented and analyzed. The analysis involves a group-based approach, in which we distinguish between three different categories of classification methods. Each category behaves differently, and the evidence obtained shows that the use of determined missing values imputation methods could improve the accuracy obtained for these methods. In this study, the convenience of using imputation methods for preprocessing data sets with missing values is stated. The analysis suggests that the use of particular imputation methods conditioned to the groups is required.

Keywords Approximate models · Classification · Imputation · Rule induction learning · Lazy learning · Missing values · Single imputation

1 Introduction

Many existing, industrial, and research data sets contain missing values (MVs). There are various reasons for their existence, such as manual data entry procedures, equipment errors, and incorrect measurements. The presence of such imperfections usually requires a preprocessing

J. Luengo (✉) · F. Herrera
Department of Computer Science and Artificial Intelligence,
CITIC-University of Granada, 18071 Granada, Spain
e-mail: julianlm@decsai.ugr.es

F. Herrera
e-mail: herrera@decsai.ugr.es

S. García
Dept. of Computer Science, University of Jaén, 23071 Jaén, Spain
e-mail: sglopez@ujaen.es

Published online: 14 June 2011

stage in which the data are prepared and cleaned [48], in order to be useful to and sufficiently clear for the knowledge extraction process. The simplest way of dealing with MVs is to discard the examples that contain them. However, this method is practical only when the data contain a relatively small number of examples with MVs and when analysis of the complete examples will not lead to serious bias during the inference [34].

MVs make the performance of data analysis difficult. The presence of MVs can also pose serious problems for researchers. In fact, inappropriate handling of the MVs in the analysis may introduce bias and can result in misleading conclusions being drawn from a research study and can also limit the generalizability of the research findings [60]. The following three types of problem are usually associated with MVs in data mining [5]: (1) loss of efficiency; (2) complications in handling and analyzing the data; and (3) bias resulting from differences between missing and complete data.

In the particular case of classification, learning from incomplete data becomes even more important. Incomplete data in either the training set or test set or in both sets affect the prediction accuracy of learned classifiers [25]. The seriousness of this problem depends in part on the proportion of MVs. Most classification algorithms cannot work directly with incomplete data sets, and due to the high dimensionality of real problems (i.e. large number of cases), it is possible that no valid (complete) cases would be present in the data set [23]. Therefore, it is important to analyze which is the best technique or preprocessing considered in order to treat the present MVs before applying the classification methods as no other option is possible.

Usually, the treatment of MVs in data mining can be handled in three different ways [18]:

- The first approach is to discard the examples with MVs in their attributes. Therefore, deleting attributes with elevated levels of MVs is included in this category too.
- Another approach is the use of maximum likelihood procedures, where the parameters of a model for the complete data are estimated, and later used for imputation by means of sampling.
- Finally, the imputation of MVs is a class of procedures that aims to fill in the MVs with estimated ones. In most cases, a data set's attributes are not independent of each other. Thus, through the identification of relationships among attributes, MVs can be determined

We will focus our attention on the use of imputation methods. A fundamental advantage of this approach is that the MV treatment is independent of the learning algorithm used. For this reason, the user can select the most appropriate method for each situation he faces. There is a wide family of imputation methods, from simple imputation techniques like mean substitution, K-nearest neighbor, etc. to those which analyze the relationships between attributes such as support vector machine-based, clustering-based, logistic regressions, maximum-likelihood procedures, and multiple imputation [6,19].

The literature on imputation methods in data mining employs well-known machine learning methods for their studies, in which the authors show the convenience of imputing the MVs for the mentioned algorithms, particularly for classification. The vast majority of MVs studies in classification usually analyze and compare one imputation method against a few others under controlled amounts of MVs and induce them artificially with known mechanisms and probability distributions [1,6,19,28,33].

We want to analyze the effect of the use of a large set of imputation methods on all the considered classifiers. Most of the considered classification methods have been used previously in MVs studies. However, they have been considered all together. In this work, we will establish three groups of classifiers to categorize them, and we will examine the best imputation strategies for each group. The former groups are as follows:

- The first group consists of the *rule induction learning* category. This group refers to algorithms that infer rules using different strategies. Therefore, we can identify as belonging to this category those methods that produce a set of more or less interpretable rules. These rules include discrete and/or continuous features, which will be treated by each method depending on their definition and representation. This type of classification methods has been the most used in case of imperfect data [49].
- The second group represents the *approximate models*. It includes artificial neural networks, support vector machines, and statistical learning. In this group, we include the methods that act like a black box. Therefore, those methods that do not produce an interpretable model fall under this category. Although the Naïve Bayes method is not a completely black box method, we have considered that this is the most appropriate category for it.
- The third and last group corresponds to the *lazy learning* category. This group includes methods that do not create any model, but use the training data to perform the classification directly. This process implies the presence of measures of similarity of some kind. Thus, the methods that use a similarity function to relate the inputs to the training set are considered as belonging to this category.

In order to perform the analysis, we use a large bunch of data sets, twenty-one in total. All the data sets have their proper MVs, and we do not induce them, as we want to stay as close to the real-world data as possible. First, we analyze the use of the different imputation strategies versus case deletion and the total lack of MVs treatment, for a total of fourteen imputation methods, for each data set. All the imputation and classification algorithms are publicly available in the KEEL software[1] [2]. These results are compared using the Wilcoxon Signed Rank test [13,24] in order to obtain the best method(s) for each classifier. With this information, we can extract the best imputation method for the three groups and indicate the best global option using a set of average rankings.

We have also analyzed two metrics related to the data characteristics, formerly known as Wilson's noise ratio and mutual information. Using these measures, we have observed the influence of the imputation procedures on the noise and on the relationship of the attributes with the class label as well. This procedure tries to quantify the quality of each imputation method independently of the classification algorithm and to observe the theoretical advantages of the imputation methods *a priori*.

The obtained results will help us to explain how imputation may be a useful tool to overcome the negative impact of MVs and the most suitable imputation method for each classifier, each group and all the classification methods together.

The rest of the paper is organized as follows. In Sect. 2, we present the basis of the application of the imputation methods, the description of the imputation methods we have used, and a brief review of the current state of the art in imputation methods. In Sect. 3, the experimental framework, the classification methods, and the parameters used for both imputation and classification methods are presented. In Sect. 4, the results obtained are analyzed. In Sect. 5, we use two measures to quantify the influence of the imputation methods in the data sets, both in the instances and in the features. Finally, in Sect. 6, we make some concluding remarks.

[1] http://keel.es.

2 Imputation background

In this section, we first set the basis of our study in accordance with the MV literature. The rest of this section is organized as follows: In Sect. 2.1, we show a brief snapshot of the latest advances in imputation methods for classification, and in Sect. 2.2, we have summarized the imputation methods that we have used in our study.

A more extensive and detailed description of these methods can be found on the web page http://sci2s.ugr.es/MVDM, and a PDF file with the original source paper descriptions is present on the web page formerly named "Imputation of Missing Values. Methods' Description". A more complete bibliography section is also available on the mentioned web page.

It is important to categorize the mechanisms, which lead to the introduction of MVs [34]. The assumptions we make about the missingness mechanism and the MVs pattern of MVs can affect which imputation method could be applied, if any. As Little and Rubin (1987) stated, there are three different mechanisms for MVs induction.

1. Missing completely at random (**MCAR**), when the distribution of an example having a missing value for an attribute does not depend on either the observed data or the MVs.
2. Missing at random (**MAR**), when the distribution of an example having a missing value for an attribute depends on the observed data, but does not depend on the MVs.
3. Not missing at random (**NMAR**), when the distribution of an example having a missing value for an attribute depends on the MVs.

In case of the MCAR mode, the assumption is that the underlying distributions of missing and complete data are the same, while for the MAR mode they are different, and the MVs can be predicted using the complete data [34]. These two mechanisms are assumed by the imputation methods so far. As Farhangfar et al. [19] and Matsubara et al. [36] state, it is only in the MCAR mechanism case where the analysis of the remaining complete data (ignoring the incomplete data) could give a valid inference (classification in our case) due to the assumption of equal distributions. That is, case and attribute removal with MVs should be applied only if the MVs are MCAR, as both of the other mechanisms could potentially lead to information loss that would lead to the generation of a biased/incorrect classifier (i.e., a classifier based on a different distribution).

Another approach is to convert the MVs to a new value (encode them into a new numerical value), but such a simplistic method was shown to lead to serious inference problems [54]. On the other hand, if a significant number of examples contain MVs for a relatively small number of attributes, it may be beneficial to perform imputation (filling-in) of the MVs. In order to do so, the assumption of MAR randomness is needed, as Little and Rubin [34] observed in their analysis.

In our case, we will use single imputation methods, due to the large time requirements of the multiple imputation schemes, and the assumptions they make regarding data distribution and MV randomness, that is, that we should know the underlying distributions of the complete data and MVs prior to their application. In addition to this, Gheyas and Smith [25] indicate that the single imputation methods are able to show better prediction capabilities than multiple imputation ones for a wide range of data sets due to the lower overfitting responses of the former ones.

2.1 An overview of the analysis of imputation methods in the literature for classification

The use of imputation methods for MVs is a task with a well-established background. It is possible to track the first formal studies to several decades ago. The work of Little and Rubin

[34] laid the foundation of further works in this topic, specially in statistics, as we have seen in the introduction to this section. From their work, imputation techniques based on sampling from estimated data distributions followed, distinguishing between single imputation procedures (like Expectation-Maximization procedures [53]) and multiple imputation ones [54], being the latter more reliable and powerful but more difficult and restrictive to be applied.

These imputation procedures became very popular for quantitative data, and therefore, they were easily adopted in other fields of knowledge, like bioinformatics [29,42,57], climatic science [55], medicine [59], etc. The imputation methods proposed in each field are adapted to the common characteristics of the data analyzed in it. With the popularization of the data mining field, many studies in the treatment of MVs arose in this topic, particularly in the classification task. Some of the existent imputation procedures of other fields are adapted to be used in classification, for example adapting them to treat with qualitative data, while many specific approaches are proposed.

In this sense, we can refer to the initial comparisons of [27] who compare the performance of the LERS classification method with the application of nine different methods for MVs: the C4.5 probability-based mechanism, the mean/mode substitution (MC, CMC), LEM2 based, Eventcovering (EC), and assigning all possible values. Their results state that the use of these imputation methods shows that, on average, imputation helps to improve classification accuracy, and the best imputation for LERS was achieved with the C4.5 internal "imputation" method. Batista and Monard [6] tested the classification accuracy of two popular classifiers (C4.5 and CN2) considering the proposal of K-NN as an imputation (KNNI) method and MC. Both CN2 and C4.5 (like [27]) algorithms have their own MV estimation. From their study, KNNI results in good accuracy, but only when the attributes are not highly correlated with each other. Related to this work, Acuna and Rodriguez [1] have investigated the effect of four methods that deal with MVs. As in [6], they use KNNI and two other imputation methods (MC and median imputation). They also use the K-NN and Linear Discriminant Analysis classifiers. The results of their study show that no significant harmful effect in accuracy is obtained from the imputation procedure. In addition to this, they state that the KNNI method is more robust in the increment in MVs in the data set in respect of the other compared methods.

The idea of using machine learning or soft computing techniques as imputation methods spreads from this point on. Li et al. [33] use a fuzzy clustering method: the Fuzzy K-means (FKMI). They compare the FKMI with mean substitution and KMI (K-means imputation). Using a Root Mean Square Error error analysis, they state that the basic KMI algorithm outperforms the MC method. Experiments also show that the overall performance of the FKMI method is better than that of the basic KMI method, particularly when the percentage of MVs is high. Feng et al. [21] use an SVM for filling in MVs (SVMI) but they do not compare this with any other imputation methods. Furthermore, they state that we should select enough complete examples where there is no MVs as the training data set in this case.

We can find more recent analysis and proposals of imputation methods, which considers an increasing number of techniques compared:

- Hruschka et al. [28] propose two imputation methods based on Bayesian networks. They compare them with 4 classical imputation methods: EM, Data Augmentation, C4.5, and the CMC method, using 4 nominal data sets from the UCI repository [3] with natural MVs (but inducing MVs in them as well). In their analysis, they employ 4 classifiers as follows: one-rule, Naïve-Bayes, C4.5, and PART. As performance measures, the authors measure the prediction value (i.e., the similarity of the imputed value to the original removed one) and the classification accuracy obtained with the four mentioned models. From the results, the authors state that better prediction results do not imply better classification results.

- Farhangfar et al. [18] take as the objective of their paper to develop a unified framework supporting a host of imputation methods. Their study inserts some imputation methods into their framework (Naïve-Bayes and Hot Deck) and compares this with other basic methods: mean, Linear Discriminant Analysis, Logreg, etc. All their experimentation is based on discrete data, so they use the "accuracy" of imputed values against randomly generated MVs. The relation of this imputation accuracy to classification accuracy is not studied. In Farhangfar et al. [19], the previous study is extended using discrete data, comparing with more classical imputation methods. This study uses a representative method of several classifiers' types as follows: decision trees, instance-based learning, rule-based classifier, probabilistic methods, and SVMs by means of boosting [51]. The MVs are produced artificially in a wide-ranging amount for each of the data sets, and the results obtained from the classification of imputed data are compared with the ones with MVs. This study shows that the impact of the imputation varies among different classifiers and that imputation is beneficial for most amounts of MVs above 5% and that the amount of improvement does not depend on the amount of MVs. The performed experimental study also shows that there is no universally best imputation method.
- Song et al. [56] study the relationship between the use of the KNNI method and the C4.5 performance (counting with its proper MV technique) over 6 data sets of software projects. They emphasize the different MVs' mechanisms (MCAR, MAR, and NMAR) and the amount of MVs introduced. From their analysis, they found results that agree with Batista and Monard [6]: KNNI can improve the C4.5 accuracy. They ran a Mann–Whitney statistical test to obtain significant differences in this statement. They also show that the missingness mechanism and pattern affect the classifier and imputation method performance.
- Twala [58] empirically analyzes 7 different procedures to treat artificial MVs for decision trees over 21 real data sets. From the study, it can be concluded that listwise deletion is the worst choice, while the multiple imputation strategy performs better than the rest of the imputation methods (particularly those with high amounts of MVs), although there is no outstanding procedure.
- García-Laencina et al. [23] evaluate the influence of imputing MVs into the classification accuracy obtained by an artificial neural network (multilayer perceptron). Four imputation techniques are considered as follows: KNNI, SOM imputation, MLP imputation, and EM over one synthetic and two real data sets, varying the amount of MVs introduced. They conclude that in real-life scenarios a detailed study is required in order to evaluate which MVs estimation can help to enhance the classification accuracy.
- Luengo et al. [35] study several imputation methods for RBFNs classifiers, both for natural and artificial (MCAR) MVs. From their results can be seen that the EC method has a good synergy with respect to the RBFN methods, as it provides better improvements in classification accuracy.
- Ding and Simonoff [14] investigate eight different missingness patterns, depending on the relationship between the missingness and three types of variables, the observed predictors, the unobserved predictors (the missing values), and the response variable. They focus on the case of classification trees for binary data (C4.5 and CART) using a modeling bankruptcy database, showing that the relationship between the missingness and the dependent variable, as well as the existence or non-existence of MVs in the testing data, is the most helpful criterion to distinguish different MVs methods.

- Gheyas and Smith [25] propose a single imputation method and a multiple imputation method, both of them based on a generalized regression neural network (GRNN). Their proposal is compared with 25 imputation methods of different natures, from machine learning methods to several variants of GRNNs. Ninty-eight data sets are used in order to introduce MVs with MCAR, MAR, and NMAR mechanisms. Then, the results of the imputation methods are compared by means of 3 different criteria, using the following three classifiers: MLP, logistic regression, and a GRNN-based classifier, showing the advantages of the proposal.

Recently, the treatment of MVs has been considered in conjunction with other hot topics in classification, like imbalanced data sets, semi-supervised learning, temporal databases, scalability, and the presence of noisy data. Nogueira et al. [41] presented a comparison of techniques used to recover values in a real imbalanced database, with a massive occurrence of MVs. This makes the process of obtaining a set of representative records, used for the recovering techniques, difficult. They used C4.5, Naïve-Bayes, K-NN, and multilayer perceptron as classifiers. To treat the MVs, they applied several techniques as follows: default value substitution or related attribute recovery. The latter tries to obtain the missing value from the information of another attribute. In addition to this, cleaning of instances/attributes with too many MVs was also carried out.

Saar-Tsechansky and Provost [52] compare several different methods (predictive value imputation, the distribution-based imputation used by C4.5 and using reduced models) for applying classification trees to instances with missing values. They distinguish between MVs in "training" (usual MVs) and MVs in "prediction" time (i.e., test partition) and adapt the novel-reduced models to this scenario. The results show that for the predictive value imputation and C4.5 distribution based, both can be preferable under different conditions. Their novel technique (reduced models) consistently outperforms the other two methods based on their experimentation.

Matsubara et al. [36] present an adaptation of a semi-supervised learning algorithm for imputation. They impute the MV using the C4.5 and Naïve-Bayes classifiers by means of a ranking aggregation to select the best examples. They compare the method with three qualitative UCI [3] data sets applying artificial MVs and perform a similar study to the one presented by Batista and Monard [6], comparing with the KNNI and MC methods. Using a non-parametric statistical test, they demonstrate the better performance of the new method over the other two in some cases.

Merlin et al. [38] propose a a new method for the determination of MVs in temporal databases based on self-organizing maps. Using two classifiers for the spatial and temporal dependencies, improvements in respect of the EM method in a hedge fund problem are shown.

We can appreciate heterogeneity from the mentioned studies. There are many different approaches for the treatment of MVs, which use many different methods (to classify and to impute MVs), but they produce similar conclusions about the convenience of using imputation methods. Therefore, in spite of the variety of studies presented, the necessity of using imputation methods for MVs is demonstrated. However, there is not an overall approximation to the selection of the best imputation technique for a wide range of classification methods.

2.2 Description of the imputation methods

In this subsection, we briefly describe the imputation methods that we have used that are the most representative and used in the literature presented in the previous subsection.

- Do Not Impute (**DNI**). As its name indicates, all the MVs remain unreplaced, so the networks must use their default MVs strategies. The objective is to verify whether imputation methods allow the classification methods to perform better than when using the original data sets. As a guideline, in [27], a previous study of imputation methods is presented.
- Case deletion or Ignore Missing (**IM**). Using this method, all instances with at least one MV are discarded from the data set.
- Global Most Common Attribute Value for Symbolic Attributes, and Global Average Value for Numerical Attributes (**MC**) [26]. This method is very simple: For nominal attributes, the MV is replaced with the most common attribute value, and numerical values are replaced with the average of all values of the corresponding attribute.
- Concept Most Common Attribute Value for Symbolic Attributes, and Concept Average Value for Numerical Attributes (**CMC**) [26]. As stated in *MC*, the MV is replaced by the most repeated one if nominal or the mean value is numerical, but considering only the instances with the same class as the reference instance.
- Imputation with K-Nearest Neighbor (**KNNI**) [6]. Using this instance-based algorithm, every time an MV is found in a current instance, KNNI computes the k nearest neighbors and a value from them is imputed. For nominal values, the most common value among all neighbors is taken, and for numerical values, the average value is used. Therefore, a proximity measure between instances is needed for it to be defined. The Euclidean distance (it is a case of a L_p norm distance) is most commonly used in the literature.
- Weighted Imputation with K-Nearest Neighbor (**WKNNI**) [57]. The weighted K-nearest neighbor method selects the instances with similar values (in terms of distance) to a considered one, so it can impute as *KNNI* does. However, the estimated value now takes into account the different distances from the neighbors, using a weighted mean or the most repeated value according to the distance.
- K-means Clustering Imputation (**KMI**) [33]. Given a set of objects, the overall objective of clustering is to divide the data set into groups based on the similarity of objects and to minimize the intra-cluster dissimilarity. KMI measures the intra-cluster dissimilarity by the addition of distances among the objects and the centroid of the cluster which they are assigned to. A cluster centroid represents the mean value of the objects in the cluster. Once the clusters have converged, the last process is to fill in all the non-reference attributes for each incomplete object based on the cluster information. Data objects that belong to the same cluster are taken to be nearest neighbors of each other, and KMI applies a nearest neighbor algorithm to replace MVs, in a similar way to KNNI.
- Imputation with Fuzzy K-means Clustering (**FKMI**) [1,33]. In fuzzy clustering, each data object has a membership function, which describes the degree to which this data object belongs to a certain cluster. In the process of updating membership functions and centroids, FKMI only takes into account complete attributes. In this process, the data object cannot be assigned to a concrete cluster represented by a cluster centroid (as is done in the basic K-mean clustering algorithm), because each data object belongs to all K clusters with different membership degrees. FKMI replaces non-reference attributes for each incomplete data object based on the information about membership degrees and the values of cluster centroids.
- Support Vector Machines Imputation (**SVMI**) [21] is an SVM regression-based algorithm to fill in MVs, i.e., set the decision attributes (output or classes) as the condition attributes (input attributes) and the condition attributes as the decision attributes, so SVM regression can be used to predict the missing condition attribute values. In order to do that, first SVMI selects the examples in which there are no missing attribute values. In

the next step, the method sets one of the condition attributes (input attribute), some of those values that are missing, as the decision attribute (output attribute), and the decision attributes as the condition attributes by contraries. Finally, an SVM regression is used to predict the decision attribute values.

- Event Covering (**EC**) [62]. Based on the work of Wong et al., a mixed-mode probability model is approximated by a discrete one. First, EC discretizes the continuous components using a minimum loss of information criterion. Treating a mixed-mode feature n-tuple as a discrete-valued one, a new statistical approach is proposed for the synthesis of knowledge based on cluster analysis. The main advantage of this method is that it does not require either scale normalization or the ordering of discrete values. By synthesizing the data into statistical knowledge, the EC method involves the following processes: (1) synthesize and detect from data inherent patterns which indicate statistical interdependency; (2) group the given data into inherent clusters based on this detected interdependency; and (3) interpret the underlying patterns for each cluster identified. The method of synthesis is based on the author's *event–covering* approach. With the developed inference method, EC is able to estimate the MVs in the data.
- Regularized Expectation-Maximization (**EM**) [55]. MVs are imputed with a regularized expectation maximization (EM) algorithm. In an iteration of the EM algorithm, given estimates of the mean and of the covariance matrix are revised in three steps. First, for each record with MVs, the regression parameters of the variables with MVs among the variables with available values are computed from the estimates of the mean and of the covariance matrix. Second, the MVs in a record are filled in with their conditional expectation values given the available values and the estimates of the mean and of the covariance matrix, the conditional expectation values being the product of the available values and the estimated regression coefficients. Third, the mean and the covariance matrix are re-estimated, the mean as the sample mean of the completed data set and the covariance matrix as the sum of the sample covariance matrix of the completed data set and an estimate of the conditional covariance matrix of the imputation error. The EM algorithm starts with initial estimates of the mean and of the covariance matrix and cycles through these steps until the imputed values and the estimates of the mean and of the covariance matrix stop changing appreciably from one iteration to the next.
- Singular Value Decomposition Imputation (**SVDI**) [57]. In this method, singular value decomposition is used to obtain a set of mutually orthogonal expression patterns that can be linearly combined to approximate the values of all attributes in the data set. In order to do that, first SVDI estimates the MVs within the *EM* algorithm, and then it computes the singular value decomposition and obtains the eigenvalues. Now, SVDI can use the eigenvalues to apply a regression to the complete attributes of the instance and to obtain an estimation of the MV itself.
- Bayesian Principal Component Analysis(**BPCA**) [42]. This method is an estimation method for MVs, which is based on Bayesian principal component analysis. Although the methodology that a probabilistic model and latent variables are estimated simultaneously within the framework of Bayesian inference is not new in principle, actual BPCA implementation that makes it possible to estimate arbitrary missing variables is new in terms of statistical methodology. The missing value estimation method based on BPCA consists of three elementary processes. They are (1) principal component (PC) regression, (2) Bayesian estimation, and (3) an expectationf́bmaximization (EM)-like repetitive algorithm.
- Local Least Squares Imputation (**LLSI**) [29]. With this method, a target instance that has MVs is represented as a linear combination of similar instances. Rather than using all

available genes in the data, only similar genes based on a similarity measure are used. The method has the "local" connotation. There are two steps in the LLSI. The first step is to select k genes by the L_2-norm. The second step is regression and estimation, regardless of how the k genes are selected. A heuristic k parameter selection method is used by the authors.

3 Experimental framework

When analyzing imputation methods, a wide range of setups can be observed. The data sets used, their type (real or synthetic), the origin, and amount of MVs, etc. must be carefully described, as the results will strongly depend on them. All these aspects are described in Sect. 3.1.

The good or bad estimations performed by the imputation method will be analyzed with regard to the accuracy obtained by many classification methods. They are presented in Sect. 3.2, grouped in the different families that we have considered, so that we can extract specialized conclusions relating the imputation results to similar classification methods.

Not all the classification methods are capable of managing the MVs on their own. It is important to indicate which methods can and the strategy that we follow when the contrary case occurs. In Sect. 3.3, we tackle the different situations that appear when using Do Not Impute.

The results obtained by the classification methods depend on the previous imputation step, but also on the parameter configuration used by both the imputation and the classification methods. Therefore, they must be indicated in order to be able to reproduce any results obtained. In Sect. 3.4, the parameter configurations used by all the methods considered in this study are presented.

3.1 Data sets description

The experimentation has been carried out using 21 benchmark data sets from the UCI repository [3]. Each data set is described by a set of characteristics such as the number of data samples, attributes, and classes, summarized in Table 1. In this table, the percentage of MVs is indicated as well: the percentage of values which are missing and the percentage of instances with at least one MV.

We have not any knowledge about the MV generation mechanism in the considered data sets. As stated in the previous analysis, it is reasonable to assume that they are distributed in an *MAR* way. Therefore, the application of the imputation methods is feasible.

Most of the previous studies in Sect. 2.1 discard any previous natural MVs, if any, and then generate random MVs for their experiments with different percentages and MV distributions.

In our study, we want to deal with the original MVs and therefore obtain the real accuracy values of each data set with the imputation methods. The amount of data sets we have used is large enough to allow us to draw significant conclusions and is larger than the majority of studies presented in Sect. 2.1. In addition to this, we use all kinds of data sets, which includes nominal data sets, numeric data sets, and data sets with mixed attributes.

In order to carry out the experimentation, we have used a 10-fold cross-validation scheme. All the classification algorithms use the same partitions, to perform fair comparisons. We take the mean accuracy of training and test of the 10 partitions as a representative measure of the method's performance.

Table 1 Data sets used

Data set	Acronym	# Instances	# Attributes	# Classes	% MV	% Inst. with MV
Cleveland	CLE	303	14	5	0.14	1.98
Wisconsin	WIS	699	10	2	0.23	2.29
Credit	CRX	689	16	2	0.61	5.37
Breast	BRE	286	10	2	0.31	3.15
Autos	AUT	205	26	6	1.11	22.44
Primary tumor	PRT	339	18	21	3.69	61.06
Dermatology	DER	365	35	6	0.06	2.19
House-votes-84	HOV	434	17	2	5.3	46.54
Water-treatment	WAT	526	39	13	2.84	27.76
Sponge	SPO	76	46	12	0.63	28.95
Bands	BAN	540	40	2	4.63	48.7
Horse-colic	HOC	368	24	2	21.82	98.1
Audiology	AUD	226	71	24	1.98	98.23
Lung-cancer	LUN	32	57	3	0.27	15.63
Hepatitis	HEP	155	20	2	5.39	48.39
Mushroom	MUS	8124	23	2	1.33	30.53
Post-operative	POS	90	9	3	0.37	3.33
Echocardiogram	ECH	132	12	4	4.73	34.09
Soybean	SOY	307	36	19	6.44	13.36
Mammographic	MAM	961	6	2	2.81	13.63
Ozone	OZO	2534	73	2	8.07	27.11

All these data sets have natural MVs, and we have imputed them with the following scheme. With the training partition, we apply the imputation method, extracting the relationships between the attributes and filling in this partition. Next, with the information obtained, we fill in the MVs in the test partition. Since we have 14 imputation methods, we will obtain 14 instances of each partition of a given data set once they have been preprocessed. All these partitions will be used to train the classification methods used in our study, and then, we will perform the test validation with the corresponding test partition. If the imputation method works only with numerical data, the nominal values are considered as a list of integer values, starting from 1 to the amount of different nominal values in the attribute.

3.2 Classification methods

In order to test the performance of the imputation methods, we have selected a set of representative classifiers. We can group them in three subcategories. In Table 2, we summarize the classification methods we have used, organized in these three categories. The description of the former categories is as follows:

- The first group is the *rule induction learning* category. This group refers to algorithms that infer rules using different strategies.
- The second group represents the *approximate models*. It includes the following: artificial neural networks, support vector machines, and statistical learning.

Table 2 Classifiers used by categories

Method	Acronym	Reference
Rule induction learning		
C4.5	C4.5	[50]
Ripper	Ripper	[10]
CN2	CN2	[9]
AQ-15	AQ	[39]
PART	PART	[22]
Slipper	Slipper	[11]
Scalable rule induction induction	SRI	[45]
Rule induction two in one	Ritio	[63]
Rule extraction system version 6	Rule-6	[44]
Approximate models		
Multilayer perceptron	MLP	[40]
C-SVM	C-SVM	[17]
ν-SVM	ν-SVM	[17]
Sequential minimal optimization	SMO	[47]
Radial basis function network	RBFN	[8]
RBFN decremental	RBFND	[8]
RBFN incremental	RBFNI	[46]
Logistic	LOG	[32]
Naïve-Bayes	NB	[15]
Learning vector quantization	LVQ	[7]
Lazy learning		
1-NN	1-NN	[37]
3-NN	3-NN	[37]
Locally weighted learning	LWL	[4]
Lazy learning of Bayesian rules	LBR	[64]

- The third and last group corresponds to the *lazy learning* category. This group incorporates methods which do not create any model but use the training data to perform the classification directly.

Many of these classifiers appear in the previous studies mentioned in Sect. 2.1. We have included an increased number of methods in our study (classical and currently most used), so we can generalize better from the obtained results.

On the other hand, some methods do not work with numerical attributes (CN2, AQ and Naïve-Bayes). In order to discretize the numerical values, we have used the well-known discretizer proposed by Fayyad and Irani [20].

For the SVM methods (C-SVM, ν-SVM, and SMO), we have applied the usual preprocessing in the literature to these methods [17]. This preprocessing consists of normalizing the numerical attributes to the [0, 1] range and binarizing the nominal attributes.

3.3 Particular missing values treatment of the classification methods

Some of the presented classification methods in the previous section have their own MVs treatment:

- C4.5 uses a probabilistic approach to handling MVs. If there are MVs in an attribute X, C4.5 uses the subset with all known values of X to calculate the information gain. Once a test based on an attribute X is chosen, C4.5 uses a probabilistic approach to partition the instances with MVs in X. If the instance has an unknown value, this instance is assigned to all partitions with different weights for each one. The weight for the partition T_i is the probability that the instance belongs to T_i. This probability is estimated to be the sum of the weights of instances in T known to satisfy the test with outcome O_i, divided by the sum of weights of the cases in T with known values in the attribute X.
- CN2 algorithm uses a rather simple imputation method to treat MVs. Every missing value is filled in with its attribute's most common known value, before calculating the entropy measure.

Therefore, when using the DNI method, both C4.5 and CN2 will use their imputation abilities to treat the MVs. Therefore, we can compare their internal MVs treatment methods against the rest of the imputation methods from Sect. 2.2.

In case of neural networks (MLP and RBFN variants), there are some interesting proposals for this case. Ennett, Frize, and Walker [16] proposed in their study to replace the MVs with "normal" values (i.e., replaced by zero). This means that the MVs do not trigger the corresponding neuron which the MV is applied to, and the network can be trained on data with MVs, and evaluate instances with MVs as well.

The previously mentioned methods can handle the MVs in case of the DNI method. On the other hand, the rest of the classification methods cannot handle the MVs. Thus, we set the training and test accuracy to zero, as the method cannot build a model or compute a distance to the test instance.

3.4 Parameters used by the imputation and classification methods

In Table 3, we show the parameters used by each imputation method described in Sect. 2.2, in cases where the method needs a parameter. Please refer to their respective papers for further descriptions of the parameters' meaning.

The values chosen are those recommended by their respective authors, as they have analyzed of the best parameter configuration in each case. Nevertheless, some indications must be made to this respect attending to the nature of the imputation method considered.

In case of neighbor-based imputation methods, i.e., KNNI, WKNNI, KMI, and FKMI, increments in the K value do not significantly improve the results but increase the amount of time needed to impute the values. This is specially critical for the FKMI method, which becomes very slow with small increments in K. The LLSI method is not very sensitive to the maximum number of neighbors, as this parameter is adjusted dynamically depending on the data set.

The EC method is also sensible to the T parameter. Incrementing it makes the imputation process faster but less clusters are created, and therefore, a worse imputation is obtained as only few reference clusters centers are used to impute. On the other hand, decrementing it will significantly increase the running time. SVMI is subject to the parameter adjusting procedure followed in SVM for classification. The EM procedure is not significantly affected by their parameters except the number of iterations. In our experimentation, we have checked that 30 iterations allow EM to converge to the stagnation limit desired. This is also relevant to the SVDI imputation method which uses EM as an initial imputation guess.

In Table 4, the parameters used by the different classification methods are presented. All these parameters are the recommended ones that have been extracted from the respective

Table 3 Methods Parameters

Method	Parameters
SVMI	Kernel = RBF
	$C = 1.0$
	Epsilon = 0.001
	Shrinking = No
KNNI, WKNNI	$K = 10$
KMI	$K = 10$
	Iterations = 100
	Error = 100
FKMI	$K = 3$
	Iterations = 100
	Error = 100
	$m = 1.5$
EC	$T = 0.05$
EM	Iterations = 30
	Stagnation tolerance = 0.0001
	Inflation factor = 1
	Regression type = multiple ridge regression
SVDI	Iterations = 30
	Stagnation tolerance = 0.005
	Inflation factor = 1
	Regression type = multiple ridge regression
	Singular vectors = 10
LLSI	Max number of nearest neighbor = 200

publications of the methods. Please refer to the associated publications listed in Table 2 to obtain the meaning of the different parameters.

4 Experimental results

In this section, we analyze the experimental results obtained. We have created an associated web page with all the results related to our analysis. The reason for this is to avoid long appendices, due to the size of the combination of all the imputation methods with the classification methods. The address is http://sci2s.ugr.es/KAIS-MVDM/. We have also included on this web page the partitions of the used data sets for further comparisons. In order to compare the algorithms and MV methods, we have used the Wilcoxon Signed Rank test, to support our analysis with a statistical test that provides us with statistical evidence of the good behavior of any approach. Therefore, the mentioned web page contains the following two documents:

- A document with all the accuracy results in both training and test for all the classification methods, each one with the 14 imputation methods.
- A document with a table summarizing the Wilcoxon test for all the imputation methods in respect of a determined classification method. The outcomes of the tables are based directly on the test accuracy results of the previous document.

Table 4 Parameters used by the classification methods

Method	Parameters
C4.5	Prune = true, confidence = 0.25, instances per leaf = 2
Ripper	Grow percentage = 0.66, $K = 2$
CN2	Percentage ex. To cover = 0.95, star size = 5, disjunt selectors = no
AQ	Star size = 5, disjunt selector = no
PART	Confidence = 0.25, intemsets per leaf = 2
Slipper	Grow percentage = 0.66, $K = 2$
SRI	Beam width = 8, min positives = 2, min negatives = 1
Ritio	–
Rule-6	Beam width = 5, min positives = 2, min negatives = 1
MLP	Hidden layers = 1, neurons per layer = 10
C-SVM	Kernel = poly., $C = 100$, eps = 0.001, degree = 1, gamma = 0.01, coef 0 = 0, $p = 1$, shrink = yes
Nu-SVM	Kernel = poly., nu = 0.1, eps = 0.001, degree = 1, gamma = 0.01, coef 0 = 0, $p = 1$, shrink = yes
SMO	C = 1,tolerance = 0.001, eps = 1e-12, Kernel = polynomial, exp = 1, lowerOrder = no
RBFN	Neurons = 50
RBFND	Percent = 0.1, initial neurons = 20, alpha = 0.3
RBFNI	Epsilon = 0.1, alpha = 0.3, delta = 0.5
LOG	Ridge = 1e-8, iteration limit = none
NB	–
LVQ	Iterations = 100, neurons = 20,alpha = 0.3,nu = 0.8
1-NN	$K = 1$, Distance function = euclidean
3-NN	$K = 3$, Distance function = euclidean
LWL	$K = 3$, Kernel function = constant
LBR	–

The rest of the analysis is organized into two parts. First, we analyze all the methods together, without differentiating the groups. This approach is similar to the previous studies on the topic. Then, we have analyzed the methods organized by the different groups, obtaining different results. Therefore, the rest of this section is organized as follows:

- In Sect. 4.1, we introduce the comparison methodology used in the subsequent subsections.
- In Sect. 4.2, we show first the global results of the imputation methods for all the classifiers together.
- In Sect. 4.3, we study the behavior of the rule induction learning classification methods.
- In Sect. 4.4, we analyze the approximate methods.
- In Sect. 4.5, we compare the results of the imputation methods for the lazy learning algorithms.
- In Sect. 4.6, we summarize the suitability and performance of the imputation methods restricted to each group. In this way, we intend to extract the best imputation method for each type of classifier and analyze whether there is any kind of relation between them.

4.1 Comparison methodology

In order to appropriately analyze the imputation and classification methods, we use the Wilcoxon tables directly from the web page. These tables provide us with an average ranking for each imputation method. The content of the tables and its interpretation are as follows:

1. We create an $n \times n$ table for each classification method. In each cell, the outcome of the Wilcoxon Signed Rank test is shown.
2. In the aforementioned tables, if the p-value obtained by the Wilcoxon tests for a pair of imputation methods is higher than our α level, formerly 0.1, then we establish that there is a *tie* in the comparison (no significant difference was found), represented by a *D*.
3. If the p-value obtained by the Wilcoxon tests is lower than our α level, formerly 0.1, then we establish that there is a *win* (represented by a *W*) or a *loss* (represented by an *L*) in the comparison. If the method presented in the row has a better ranking than the method presented in the column in the Wilcoxon test, then there is a *win*, otherwise there is a *loss*.

With these columns, we have produced an average ranking for each classifier. We have computed the number of times that an imputation methods wins and the number of times that an imputation method wins and ties. Then, we obtain the average ranking by putting those imputation methods which have a higher "wins + ties" sum first among the rest of the imputation methods. If a draw is found for "wins + ties", we use the "wins" to establish the rank. If some methods obtain a draw for both "wins + ties" and "wins", then an average ranking is assigned to all of them.

In order to compare the imputation methods for the classification methods considered in each situation (global or family case), we have added two more final columns in the tables contained in the next subsections. In the first new column, we compute the mean of the rankings for each imputation method across all the classifiers of the correspondent group (column "Avg."), that is, the mean of every row. By doing so, we can obtain a new rank (final column RANKS), in which we propose a new ordering for the imputation methods for a given classifier's group, using the values of the column "Avg." to sort the imputation methods.

4.2 Results for all the classification methods

In this section, we analyze the different imputation approaches for all the imputation methods as a first attempt to obtain an "overall best" imputation method. Following the indications given in the previous subsection, in Table 5, the obtained average ranks and final imputation methods' rankings can be seen.

When comparing all the classifiers together, we find that it is difficult to establish differences between the imputation methods and to select the best one. The FKMI method obtains the best final ranking. However, the EC method has a very similar average ranking (5.70 for EC, 5.26 for FKMI). There are some additional methods that obtain a very similar average ranking, and they are not far from FKMI and EC. SVMI, KMI, MC, and CMC have an average ranking between 6.09 and 6.28. Therefore, we cannot firmly establish one best method from among all of them, and in this initial case, we must consider a range of good possible imputation methods for the treatment of the MVs from the mentioned ones.

We can relate the obtained results with the previous studies that consider together some of the imputation and classification methods analyzed here. Farhangfar et al. [19] conclude that imputation is beneficial for K-NN, SVM, and Ripper classifiers and that C4.5 and NB

Table 5 Average ranks for all the classifiers

	RBFN	RBFND	RBFNI	C4.5	1-NN	LOG	LVQ	MLP	NB	ν-SVM	C-SVM	Ripper	PART	Slipper	3-NN	AQ	CN2	SMO	LBR	LWL	SRI	Ritio	Rule-6	Avg.	RANKS
IM	9	6.5	4.5	5	5	6	3.5	13	12	10	5.5	8.5	1	4	11	6.5	10	5.5	5	8	6.5	6	5	6.83	7
EC	1	1	1	2.5	9.5	3	7	8.5	10	13	1	8.5	6.5	1	13	6.5	5.5	2	9	8	6.5	6	1	5.70	2
KNNI	5	6.5	10.5	9	2.5	9	7	11	6.5	8	5.5	2.5	6.5	11	5.5	11	5.5	5.5	9	8	11.5	11	11	7.76	10
WKNNI	13	6.5	4.5	11	4	10	10	4.5	6.5	4.5	5.5	2.5	6.5	7	5.5	6.5	1	5.5	9	8	11.5	6	11	6.96	8
KMI	3.5	2	7	5	12	3	11	3	4.5	8	5.5	2.5	6.5	3	5.5	6.5	5.5	9	9	2.5	9.5	12	7.5	6.24	5
FKMI	12	6.5	10.5	7.5	6	3	1.5	4.5	11	4.5	5.5	2.5	6.5	10	1.5	2	5.5	3	9	2.5	1	2	3	5.26	1
SVMI	2	11.5	2.5	1	9.5	7.5	3.5	1.5	13	8	11	5.5	6.5	7	9	1	5.5	9	3	8	6.5	6	2	6.09	3
EM	3.5	6.5	13	13	11	12	12.5	10	4.5	4.5	10	12	6.5	7	5.5	12	13	11.5	9	2.5	3	6	4	8.37	11
SVDI	9	6.5	7	11	13	11	12.5	8.5	3	11.5	12	11	6.5	12	12	10	12	11.5	1	12	9.5	10	11	9.72	12
BPCA	14	14	14	14	14	13	7	14	2	2	13	13	13	7	14	13	14	13	13	13	13	13	13	11.87	14
LLSI	6	6.5	10.5	11	7.5	7.5	7	6.5	9	4.5	5.5	5.5	6.5	7	5.5	6.5	11	9	9	8	3	6	7.5	7.22	9
MC	9	6.5	10.5	7.5	7.5	3	7	6.5	8	11.5	5.5	8.5	6.5	2	1.5	6.5	5.5	5.5	3	2.5	3	6	7.5	6.11	4
CMC	9	13	2.5	5	1	3	1.5	1.5	14	14	5.5	8.5	12	13	5.5	3	5.5	1	3	8	6.5	1	7.5	6.28	6
DNI	9	11.5	7	2.5	2.5	14	14	12	1	1	14	14	14	14	10	14	5.5	14	14	14	14	14	14	10.61	13

are robust against MVs using only nominal data. Our experiments show this fact as well, considering numerical and mixed data as well (and that C4.5 behaves better with SVMI as we have aforementioned). Although they stress that MC and CMC are the less useful approaches, our results indicate that in case of K-NN the contrary is verified when analyzing more data sets. On the other hand, Hruschka et al. [28] obtain that the best imputation method depends on the data set for C4.5, PART, and NB. However, due to the limited number of data sets used, these results are not as general as ours.

It is important to point out that the DNI and IM methods do not obtain a good rank. In particular, the DNI method obtains a very high ranking (10.61) only exceeded by the BPCA imputation method which performs very badly. The IM method has an average rank, and it is situated in the middle of the ranking. Thus, we can consider discarding the examples with MVs, or not processing them, to be inadvisable, as expected from the results presented in the previous studies.

Table 6 Average ranks for the rule induction learning methods

	C45	Ripper	PART	Slipper	AQ	CN2	SRI	Ritio	Rules-6	Avg.	RANKS
IM	5	8.5	1	4	6.5	10	6.5	6	5	5.83	4
EC	2.5	8.5	6.5	1	6.5	5.5	6.5	6	1	4.89	3
KNNI	9	2.5	6.5	11	11	5.5	11.5	11	11	8.78	11
WKNNI	11	2.5	6.5	7	6.5	1	11.5	6	11	7.00	8
KMI	5	2.5	6.5	3	6.5	5.5	9.5	12	7.5	6.44	6
FKMI	7.5	2.5	6.5	10	2	5.5	1	2	3	4.44	1
SVMI	1	5.5	6.5	7	1	5.5	6.5	6	2	4.56	2
EM	13	12	6.5	7	12	13	3	6	4	8.50	10
SVDI	11	11	6.5	12	10	12	9.5	10	11	10.33	12
BPCA	14	13	13	7	13	14	13	13	13	12.56	14
LLSI	11	5.5	6.5	7	6.5	11	3	6	7.5	7.11	9
MC	7.5	8.5	6.5	2	6.5	5.5	3	6	7.5	5.89	5
CMC	5	8.5	12	13	3	5.5	6.5	1	7.5	6.89	7
DNI	2.5	14	14	14	14	5.5	14	14	14	11.78	13

We must point out that the results obtained by the IM method should be considered with caution. Since several instances are discarded, the test and training partitions tend to be smaller than the original ones. This allows the classifiers to obtain better results in training, since there are less instances and less noise from the MVs. In tests, the classifier can achieve better results for some data sets if the remaining instances are well separated in the feature space, since a hit in the test partition counts for more in accuracy than in the other imputation methods (with complete test partitions).

From these results, it is clear that we need to reduce the amount of classifiers when trying to obtain the best imputation method. In the following subsections, we have focused on the different types of classification methods in order to avoid the high ranking variation observed in Table 5.

4.3 Results for the rule induction learning methods

In this section, we present the results of the rule induction classification methods. In Table 6, we show the ranking for each classification method belonging to this group. This table's structure is the same as that described in Sect. 4.1. Therefore, we only perform the average between the rankings obtained for the classification algorithms belonging to this group.

We can observe that, for the rule induction learning classifiers, the imputation methods FKMI, SVMI, and EC perform best. The differences between these three methods in average rankings are low. Thus, we can consider that these three imputation methods are the most suitable for this kind of classifier. They are well separated from the other imputation methods, and we cannot choose a best method from among these three. This is in contrast to the global results presented in Sect. 4.2, where little differences could be found among the first ranked imputation methods. Both FKMI and EC methods were also considered among the best in the global first approach presented in the previous subsection. Initially, Batista and Monard [6] indicated that C4.5 and CN2 benefit from the use of KNNI imputation method. However, it can be seen that C4.5 behaves better with SVMI and EC imputation methods when more data sets are considered, while CN2 works better with WKNNI.

On the other hand, BPCA and DNI are the worst methods. The BPCA method usually performs badly for all the classifiers. As DNI is also a bad option, this means that the rule

Table 7 Average ranks for the approximate methods

	RBFN	RBFND	RBFNI	LOG	LVQ	MLP	NB	ν-SVM	C-SVM	SMO	Avg.	RANKS
IM	9	6.5	4.5	6	3.5	13	12	10	5.5	5.5	7.55	10
EC	1	1	1	3	7	8.5	10	13	1	2	4.75	1
KNNI	5	6.5	10.5	9	7	11	6.5	8	5.5	5.5	7.45	9
WKNNI	13	6.5	4.5	10	10	4.5	6.5	4.5	5.5	5.5	7.05	6
KMI	3.5	2	7	3	11	3	4.5	8	5.5	9	5.65	2
FKMI	12	6.5	10.5	3	1.5	4.5	11	4.5	5.5	3	6.20	3
SVMI	2	11.5	2.5	7.5	3.5	1.5	13	8	11	9	6.95	5
EM	3.5	6.5	13	12	12.5	10	4.5	4.5	10	11.5	8.80	11
SVDI	9	6.5	7	11	12.5	8.5	3	11.5	12	11.5	9.25	12
BPCA	14	14	14	13	7	14	2	2	13	13	10.60	14
LLSI	6	6.5	10.5	7.5	7	6.5	9	4.5	5.5	9	7.20	7
MC	9	6.5	10.5	3	7	6.5	8	11.5	5.5	5.5	7.30	8
CMC	9	13	2.5	3	1.5	1.5	14	14	5.5	1	6.50	4
DNI	9	11.5	7	14	14	12	1	1	14	14	9.75	13

induction learning algorithms would greatly benefit from the use of the imputation methods, despite some of them being capable of dealing with MVs on their own.

The rest of the imputation methods spans between an average rank of 5.8 to 9, with a great difference between BPCA and DNI methods in ranking. The IM method is fourth, which could mean that the rule induction learning algorithms perform better with complete instances in training and test, but they do not work well with test instances with imputed MVs. However, avoiding test cases with MVs is not always possible.

4.4 Results for the approximate methods

In this section, we present the obtained results for the approximate models. In Table 7, we can observe the rankings associated with the methods belonging to this group. Again, this table structure is the same as described in Sect. 4.1.

In case of the approximate models, the differences between imputation methods are even more evident. We can select the EC method as the best solution, as it has a difference of ranking of almost 1 with KMI, which stands as the second best. This difference increases when considering the third best, FKMI. No other family of classifiers present this gap in the rankings. Therefore, in this family of classification methods, we could, with some confidence, establish the EC method as the best choice. This is in contrast to the global results, from which there is no outstanding method.

In [35], the analysis of several imputation methods with respect to RBFNs considering data set with both natural and induced MVs yielded that EC was the best imputation method. Therefore, these results can be extended to a larger number of approximate classification methods.

The DNI and IM methods are among the worst. This means that for the approximate methods the use of some kind of MV treatment is mandatory, whereas the EC method is the most suitable one. As with the rule induction learning methods, the BPCA method is the worst choice, with the highest ranking.

Table 8 Average ranks for the lazy learning methods

	1-NN	3-NN	LBR	LWL	Avg.	RANKS
IM	5	11	5	8	7.25	7
EC	9.5	13	9	8	9.88	12
KNNI	2.5	5.5	9	8	6.25	4
WKNNI	4	5.5	9	8	6.63	5
KMI	12	5.5	9	2.5	7.25	8
FKMI	6	1.5	9	2.5	4.75	3
SVMI	9.5	9	3	8	7.38	9
EM	11	5.5	9	2.5	7.00	6
SVDI	13	12	1	12	9.50	11
BPCA	14	14	13	13	13.50	14
LLSI	7.5	5.5	9	8	7.50	10
MC	7.5	1.5	3	2.5	3.63	1
CMC	1	5.5	3	8	4.38	2
DNI	2.5	10	14	14	10.13	13

4.5 Results for the lazy learning methods

The results for the last group are presented in Table 8. Again, this table structure is the same as described in Sect. 4.1.

For the lazy learning models, the MC method is the best with the lowest average ranking. The CMC method, which is relatively similar to MC, also obtains a low rank very close to MC's. Only the FKMI method obtains a low enough rank to be compared with the MC and CMC methods. The rest of the imputation methods is far from these lowest ranks with almost two points of difference in the ranking. This situation is similar to the rule induction learning methods' family, in which we could find three outstanding methods with a difference of 1 between the third and fourth ones. On the other hand, Acuna and Rodriguez [1] analyzed the benefits of KNNI for K-NN classifiers with respect to IM. In our experimentations, this benefit is clear, but considering additional imputation methods it can be seen that CMC and MC imputation methods are the best for 1-NN and 3-NN.

Again, the DNI and IM methods obtain high rankings. The DNI method is one of the worst, with only the BPCA method performing worse. As with the approximate models, the imputation methods produce a significant improvement in the accuracy of these classification methods and they should always be considered prior to their application.

4.6 Summary of the group-based results

In the previous Sect. 4.3, 4.4, and 4.5, we have observed that when comparing the imputation methods to similar classifiers, more significant information about the best ones can be extracted. In this section, we summarize these best imputation methods for each group, and we analyze the similarity between them. For the Wilcoxon tables with their rankings from Sects. 4.3 to 4.5, we have built Table 9 with the best three methods of each group. We have stressed in bold those rankings equal to or below three.

From Table 9, we can observe some interesting aspects:

- The rule induction learning category and the approximate models share the EC and FKMI methods in their top 3 best imputation algorithms.

Table 9 Best imputation methods for each group

	Ranking		
	Rule I. learning	Approx. models	Lazy L. models
EC	**3**	**1**	12
KMI	6	**2**	8
FKMI	**1**	**3**	**3**
SVMI	**2**	5	9
MC	5	8	**1**
CMC	7	4	**2**

The three best rankings per column are stressed in bold

- The lazy learning models only share the FKMI method in common with the rest. This means that the best method obtained in the general analysis in Sect. 4.2 is the only one present in all of the three groups as one of the best.
- The CMC and MC methods do not perform outstandingly in the rule induction learning methods and approximate models. Thus, we can consider that such a simple imputation strategy is only useful when we do not have to build any model from the data.

From these results, we can finally point out a set of imputation methods for each group that represent the best option(s) for them, that is:

- When using a rule induction learning classification method, the FKMI, SVMI, and EC imputation methods are the best choices.
- In case of the approximate models, the EC method is the best option with considerable difference. KMI and FKMI can be good choices as well.
- The lazy learning models benefit most from simple approaches as MC and CMC methods.

Therefore, we can establish that the consideration of different imputation methods is required in each case. Notice that the results for all of the three groups do not correspond to the global result from the previous section. If no information about the type of classification method is available, the FKMI imputation method is a good option no matter the classification method chosen. The consideration of the EC algorithm is advisable in this case as well.

It is also important to remark that DNI and IM methods are never the best or among the best imputation methods for any group. Only in case of the rule induction learning methods does the IM imputation method obtain a relatively low rank (4th place) as we have previously mentioned. This fact indicates that the imputation methods usually outperform the non-imputation strategies.

As a final remark, we can state that the results obtained in our study cohere with those mentioned in Sect. 2.1 and particularly with Acuna and Rodriguez [1], Batista and Monard [6], Farhangfar et al. [19], Feng 25 et al. [21], García-Laencina et al. [23], Twala [58], and Li et al. [33], that is

- The imputation methods that fill in the MVs outperform the case deletion (IM method) and the lack of imputation (DNI method).
- There is no universal imputation method that performs best for all classifiers.

Please note that we have tackled the second point by adding a categorization and a wide benchmark bed, obtaining a group of recommended imputation methods for each family.

5 Influence of the imputation on the instances and individual features

In the previous section, we have analyzed the relationship between the use of several imputation methods with respect to the classifiers' accuracy. However, it would be interesting to relate the influence of the imputation methods to the performance obtained by the classification method based on the information contained in the data set. In order to study this influence and the benefits/drawbacks of using the different imputation methods, we have considered the use of two different measures. They are described as follows:

- Wilson's Noise Ratio: This measure proposed by Wilson [61] observes the noise in the data set. For each instance of interest, the method looks for the K nearest neighbors (using the Euclidean distance) and uses the class labels of such neighbors in order to classify the considered instance. If the instance is not correctly classified, then the variable $noise$ is increased by one unit. Therefore, the final noise ratio will be

  $$\text{Wilson's Noise} = \frac{\text{noise}}{\text{\# instances in the data set}}$$

  In particular, we only compute the noise for the *imputed* instances considering $K = 5$.
- Mutual information: Mutual information (MI) is considered to be a good indicator of relevance between two random variables [12]. Recently, the use of the MI measure in feature selection has become well known and seen to be successful [31,30,43]. The use of the MI measure for continuous attributes has been tackled by [30], allowing us to compute the MI measure not only in nominal-valued data sets.

  In our approach, we calculate the MI between each input attribute and the class attribute, obtaining a set of values, one for each input attribute. In the next step, we compute the ratio between each one of these values, considering the imputation of the data set with one imputation method in respect of the not imputed data set. The average of these ratios will show us if the imputation of the data set produces a gain in information:

  $$\text{Avg. MI Ratio} = \frac{\sum_{x_i \in X} \frac{MI_\alpha(x_i)+1}{MI(x_i)+1}}{|X|}$$

  where X is the set of input attributes, $MI_\alpha(i)$ represents the MI value of the ith attribute in the imputed data set, and $MI(i)$ is the MI value of the ith input attribute in the not imputed data set. We have also applied the Laplace correction, summing 1 to both numerator and denominator, as an MI value of zero is possible for some input attributes.

  The calculation of $MI(x_i)$ depends on the type of attribute x_i. If the attribute x_i is nominal, the MI between x_i and the class label Y is computed as follows:

  $$MI_{\text{nominal}}(x_i) = I(x_i; Y) = \sum_{z \in x_i} \sum_{y \in Y} p(z, y) \log_2 \frac{p(z, y)}{p(z)p(y)}.$$

  On the other hand, if the attribute x_i is numeric, we have used the Parzen window density estimate as shown in [30] considering a Gaussian window function:

  $$MI_{\text{numeric}}(x_i) = I(x_i; Y) = H(Y) - H(C|X);$$

  where $H(Y)$ is the entropy of the class label

  $$H(Y) = -\sum_{y \in Y} p(y) \log_2 p(y);$$

and $H(C|X)$ is the conditional entropy

$$H(Y|x_i) = -\sum_{z \in x_i} \sum_{y \in Y} p(z, y) \log_2 p(y|z).$$

Considering that each sample has the same probability, applying the Bayesian rule and approximating $p(y|z)$ by the Parzen window we get:

$$\hat{H}(Y|x_i) = -\sum_{j=1}^{n} \frac{1}{n} \sum_{y=1}^{N} \hat{p}(y|z_j) \log_2 \hat{p}(y|z_j)$$

where n is the number of instances in the data set, N is the total number of class labels, and $\hat{p}(c|x)$ is

$$\hat{p}(y|z) = \frac{\sum_{i \in I_c} \exp\left(-\frac{(z-z_i)\Sigma^{-1}(z-z_i)}{2h^2}\right)}{\sum_{k=1}^{N} \sum_{i \in I_k} \exp\left(-\frac{(z-z_i)\Sigma^{-1}(z-z_i)}{2h^2}\right)}.$$

In this case, I_c is the set of indices of the training examples belonging to class c, and Σ is the covariance of the random variable $(z - z_i)$.

Comparing with Wilson's noise ratio, we can observe which imputation methods reduce the impact of the MVs as a noise and which methods produce noise when imputing. In addition, the MI ratio allows us to relate the attributes to the imputation results. A value of the MI ratio higher than 1 will indicate that the imputation is capable of relating more of the attributes individually to the class labels. A value lower than 1 will indicate that the imputation method is adversely affecting the relationship between the individual attributes and the class label.

In Table 10, we have summarized the Wilson's noise ratio values for the 21 data sets considered in our study. We must point out that the results of Wilson's noise ratio are related to a given data set. Hence, the characteristics of the proper data appear to determine the values of this measure.

In Table 11, we have summarized the average MI ratios for the 21 data sets. In the results, we can observe that the average ratios are usually close to 1; that is, the use of imputation methods appears to harm the relationship between the class label and the input attribute little or not at all, even improving it in some cases. However, the mutual information considers only one attribute at a time, and therefore, the relationships between the input attributes are ignored. The imputation methods estimate the MVs using such relationships and can afford improvements in the performance of the classifiers. Hence, the highest values of average MI ratios could be related to those methods which can obtain better estimates for the MVs and maintaining the relationship degree between the class labels and the isolated input attributes. It is interesting to note that when analyzing the MI ratio, the values do not appear to be as highly data dependant as Wilson's noise ratio, as the values for all the data sets are more or less close to each other.

If we count the methods with the lowest Wilson's noise ratios in each data set in Table 10, we find that the CMC method is first, with 12 times the lowest one, and the EC method is second with 9 times the lowest one. If we count the methods with the highest mutual information ratio in each data set in Table 11, the EC method has the highest ratio for 7 data sets and is therefore the first one. The CMC method has the highest ratio for 5 data sets and is the second one in this case. Considering the analysis of the previous Sect. 4.6 with these two methods:

Table 10 Wilson's noise ratio values

Data set	Imp. method	% Wilson's Noise ratio	Data set	Imp. method	% Wilson's Noise ratio	Data set	Imp. method	% Wilson's Noise ratio
CLE	MC	50.0000	HOV	MC	7.9208	HEP	MC	17.3333
	CMC	50.0000		CMC	5.4455		CMC	**16.0000**
	KNNI	50.0000		KNNI	7.4257		KNNI	20.0000
	WKNNI	50.0000		WKNNI	7.4257		WKNNI	20.0000
	KMI	50.0000		KMI	7.4257		KMI	20.0000
	FKMI	50.0000		FKMI	7.9208		FKMI	17.3333
	SVMI	50.0000		SVMI	6.9307		SVMI	17.3333
	EM	66.6667		EM	11.8812		EM	22.6667
	SVDI	66.6667		SVDI	8.9109		SVDI	21.3333
	BPCA	50.0000		BPCA	6.9307		BPCA	21.3333
	LLSI	50.0000		LLSI	**4.9505**		LLSI	18.6667
	EC	**33.3333**		EC	7.4257		EC	**16.0000**
WIS	MC	18.7500	WAT	MC	31.5068	MUS	MC	**0.0000**
	CMC	**12.5000**		CMC	**21.2329**		CMC	**0.0000**
	KNNI	**12.5000**		KNNI	27.3973		KNNI	**0.0000**
	WKNNI	**12.5000**		WKNNI	27.3973		WKNNI	**0.0000**
	KMI	**12.5000**		KMI	27.3973		KMI	**0.0000**
	FKMI	**12.5000**		FKMI	31.5068		FKMI	**0.0000**
	SVMI	**12.5000**		SVMI	23.9726		SVMI	**0.0000**
	EM	**12.5000**		EM	46.5753		EM	**0.0000**
	SVDI	**12.5000**		SVDI	49.3151		SVDI	**0.0000**
	BPCA	**12.5000**		BPCA	26.0274		BPCA	**0.0000**
	LLSI	**12.5000**		LLSI	25.3425		LLSI	**0.0000**
	EC	**12.5000**		EC	22.6027		EC	**0.0000**
CRX	MC	18.9189	SPO	MC	27.2727	POS	MC	**33.3333**
	CMC	18.9189		CMC	**22.7273**		CMC	**33.3333**
	KNNI	21.6216		KNNI	27.2727		KNNI	**33.3333**
	WKNNI	21.6216		WKNNI	27.2727		WKNNI	**33.3333**
	KMI	21.6216		KMI	27.2727		KMI	**33.3333**
	FKMI	18.9189		FKMI	27.2727		FKMI	**33.3333**
	SVMI	13.5135		SVMI	27.2727		SVMI	**33.3333**
	EM	32.4324		EM	36.3636		EM	**33.3333**
	SVDI	27.0270		SVDI	31.8182		SVDI	**33.3333**
	BPCA	21.6216		BPCA	27.2727		BPCA	**33.3333**
	LLSI	18.9189		LLSI	27.2727		LLSI	**33.3333**
	EC	**13.5135**		EC	27.2727		EC	**33.3333**
BRE	MC	55.5556	BAN	MC	25.4753	ECH	MC	40.0000
	CMC	55.5556		CMC	24.3346		CMC	40.0000
	KNNI	55.5556		KNNI	23.1939		KNNI	46.6667
	WKNNI	55.5556		WKNNI	22.8137		WKNNI	44.4444
	KMI	55.5556		KMI	25.4753		KMI	46.6667
	FKMI	55.5556		FKMI	24.3346		FKMI	40.0000

Table 10 continued

Data set	Imp. method	% Wilson's Noise ratio	Data set	Imp. method	% Wilson's Noise ratio	Data set	Imp. method	% Wilson's Noise ratio
	SVMI	55.5556		SVMI	**21.2928**		SVMI	44.4444
	EM	**44.4444**		EM	26.2357		EM	51.1111
	SVDI	**44.4444**		SVDI	22.4335		SVDI	48.8889
	BPCA	66.6667		BPCA	23.9544		BPCA	44.4444
	LLSI	66.6667		LLSI	24.7148		LLSI	**37.7778**
	EC	66.6667		EC	23.5741		EC	48.8889
AUT	MC	45.6522	HOC	MC	19.3906	SOY	MC	**2.4390**
	CMC	41.3043		CMC	**10.2493**		CMC	**2.4390**
	KNNI	41.3043		KNNI	20.2216		KNNI	**2.4390**
	WKNNI	41.3043		WKNNI	19.1136		WKNNI	**2.4390**
	KMI	41.3043		KMI	21.8837		KMI	**2.4390**
	FKMI	45.6522		FKMI	20.4986		FKMI	**2.4390**
	SVMI	43.4783		SVMI	20.2216		SVMI	**2.4390**
	EM	58.6957		EM	21.0526		EM	**2.4390**
	SVDI	52.1739		SVDI	21.0526		SVDI	7.3171
	BPCA	43.4783		BPCA	19.3906		BPCA	7.3171
	LLSI	45.6522		LLSI	20.4986		LLSI	**2.4390**
	EC	**30.4348**		EC	20.7756		EC	**2.4390**
PRT	MC	71.0145	AUD	MC	38.7387	MAM	MC	21.3740
	CMC	**60.8696**		CMC	**32.8829**		CMC	**13.7405**
	KNNI	69.5652		KNNI	38.7387		KNNI	25.9542
	WKNNI	69.5652		WKNNI	38.7387		WKNNI	25.9542
	KMI	71.0145		KMI	38.7387		KMI	24.4275
	FKMI	71.0145		FKMI	38.7387		FKMI	20.6107
	SVMI	68.1159		SVMI	37.8378		SVMI	16.7939
	EM	88.4058		EM	53.6036		EM	20.6107
	SVDI	91.7874		SVDI	46.3964		SVDI	27.4809
	BPCA	71.4976		BPCA	40.5405		BPCA	25.1908
	LLSI	69.5652		LLSI	36.9369		LLSI	26.7176
	EC	66.1836		EC	37.8378		EC	18.3206
DER	MC	**0.0000**	LUN	MC	80.0000	OZO	MC	4.8035
	CMC	**0.0000**		CMC	80.0000		CMC	**3.6390**
	KNNI	**0.0000**		KNNI	80.0000		KNNI	4.3668
	WKNNI	**0.0000**		WKNNI	80.0000		WKNNI	4.5124
	KMI	**0.0000**		KMI	80.0000		KMI	4.9491
	FKMI	**0.0000**		FKMI	80.0000		FKMI	4.0757
	SVMI	**0.0000**		SVMI	80.0000		SVMI	3.7846
	EM	**0.0000**		EM	**20.0000**		EM	4.8035
	SVDI	**0.0000**		SVDI	40.0000		SVDI	4.8035
	BPCA	**0.0000**		BPCA	80.0000		BPCA	4.3668
	LLSI	**0.0000**		LLSI	80.0000		LLSI	4.2213
	EC	**0.0000**		EC	80.0000		EC	4.8035

The best ranking per data set is stressed in bold

Table 11 Average mutual information ratio

Data set	Imp. method	Avg. MI ratio	Data set	Imp. method	Avg. MI ratio	Data set	Imp. method	Avg. MI ratio
CLE	MC	0.998195	HOV	MC	0.961834	HEP	MC	0.963765
	CMC	0.998585		CMC	1.105778		CMC	0.990694
	KNNI	0.998755		KNNI	0.965069		KNNI	0.978564
	WKNNI	0.998795		WKNNI	0.965069		WKNNI	0.978343
	KMI	0.998798		KMI	0.961525		KMI	0.980094
	FKMI	0.998889		FKMI	0.961834		FKMI	0.963476
	SVMI	0.998365		SVMI	0.908067		SVMI	1.006819
	EM	0.998152		EM	0.891668		EM	0.974433
	SVDI	0.997152		SVDI	0.850361		SVDI	0.967673
	BPCA	0.998701		BPCA	1.091675		BPCA	0.994420
	LLSI	0.998882		LLSI	**1.122904**		LLSI	0.995464
	EC	**1.000148**		EC	1.007843		EC	**1.024019**
WIS	MC	0.999004	WAT	MC	0.959488	MUS	MC	1.018382
	CMC	0.999861		CMC	0.967967		CMC	1.018382
	KNNI	0.999205		KNNI	0.961601		KNNI	0.981261
	WKNNI	0.999205		WKNNI	0.961574		WKNNI	0.981261
	KMI	0.999322		KMI	0.961361		KMI	1.018382
	FKMI	0.998923		FKMI	0.961590		FKMI	1.018382
	SVMI	0.999412		SVMI	0.967356		SVMI	0.981261
	EM	0.990030		EM	0.933846		EM	**1.142177**
	SVDI	0.987066		SVDI	0.933040		SVDI	1.137152
	BPCA	0.998951		BPCA	0.964255		BPCA	0.987472
	LLSI	0.999580		LLSI	0.964063		LLSI	0.977275
	EC	**1.000030**		EC	**1.027369**		EC	1.017366
CRX	MC	1.000883	SPO	MC	0.997675	POS	MC	1.012293
	CMC	1.000966		CMC	**1.022247**		CMC	1.012293
	KNNI	0.998823		KNNI	0.999041		KNNI	1.012293
	WKNNI	0.998870		WKNNI	0.999041		WKNNI	1.012293
	KMI	1.001760		KMI	0.998464		KMI	1.012293
	FKMI	1.000637		FKMI	0.997675		FKMI	1.012293
	SVMI	0.981878		SVMI	1.015835		SVMI	1.012293
	EM	0.985609		EM	0.982325		EM	1.012293
	SVDI	0.976398		SVDI	0.979187		SVDI	1.014698
	BPCA	0.999934		BPCA	1.006236		BPCA	1.012293
	LLSI	1.001594		LLSI	1.004821		LLSI	**1.018007**
	EC	**1.008718**		EC	1.018620		EC	0.997034
BRE	MC	0.998709	BAN	MC	1.012922	ECH	MC	0.981673
	CMC	0.998709		CMC	1.070857		CMC	0.995886
	KNNI	0.992184		KNNI	0.940369		KNNI	0.997912
	WKNNI	0.992184		WKNNI	0.940469		WKNNI	**0.998134**
	KMI	0.998709		KMI	1.016101		KMI	0.967169
	FKMI	0.998709		FKMI	1.020989		FKMI	0.983606

Table 11 continued

Data set	Imp. method	Avg. MI ratio	Data set	Imp. method	Avg. MI ratio	Data set	Imp. method	Avg. MI ratio
	SVMI	0.998709		SVMI	**1.542536**		SVMI	0.987678
	EM	**1.013758**		EM	1.350315		EM	0.967861
	SVDI	0.999089		SVDI	1.365572		SVDI	0.935855
	BPCA	1.000201		BPCA	1.010596		BPCA	0.972327
	LLSI	1.000201		LLSI	1.015033		LLSI	0.988591
	EC	1.001143		EC	1.102328		EC	0.970029
AUT	MC	0.985610	HOC	MC	0.848649	SOY	MC	1.056652
	CMC	0.991113		CMC	**2.039992**		CMC	1.123636
	KNNI	0.986239		KNNI	0.834734		KNNI	1.115818
	WKNNI	0.985953		WKNNI	0.833982		WKNNI	1.115818
	KMI	0.985602		KMI	0.821936		KMI	1.056652
	FKMI	0.984694		FKMI	0.849141		FKMI	1.056652
	SVMI	0.991850		SVMI	0.843456		SVMI	**1.772589**
	EM	0.970557		EM	0.775773		EM	1.099286
	SVDI	0.968938		SVDI	0.750930		SVDI	1.065865
	BPCA	0.986631		BPCA	0.964587		BPCA	1.121603
	LLSI	0.985362		LLSI	0.926068		LLSI	1.159610
	EC	**1.007652**		EC	0.911543		EC	1.222631
PRT	MC	0.949896	AUD	MC	0.990711	MAM	MC	0.974436
	CMC	**1.120006**		CMC	1.032162		CMC	1.029154
	KNNI	0.976351		KNNI	0.993246		KNNI	0.965926
	WKNNI	0.976351		WKNNI	0.993246		WKNNI	0.965926
	KMI	0.949896		KMI	1.000235		KMI	0.966885
	FKMI	0.949896		FKMI	0.990711		FKMI	0.974228
	SVMI	1.038152		SVMI	1.007958		SVMI	**1.272993**
	EM	0.461600		EM	1.129168		EM	0.980865
	SVDI	0.485682		SVDI	1.065091		SVDI	1.052790
	BPCA	0.987598		BPCA	1.156676		BPCA	0.978209
	LLSI	1.016230		LLSI	1.061197		LLSI	0.994349
	EC	1.053185		EC	**1.209608**		EC	1.269505
DER	MC	1.000581	LUN	MC	0.996176	OZO	MC	0.982873
	CMC	**1.002406**		CMC	1.008333		CMC	**0.989156**
	KNNI	0.999734		KNNI	0.996176		KNNI	0.982759
	WKNNI	0.999734		WKNNI	0.996176		WKNNI	0.982721
	KMI	1.000581		KMI	0.996176		KMI	0.982495
	FKMI	1.000581		FKMI	0.996176		FKMI	0.982951
	SVMI	1.001566		SVMI	1.006028		SVMI	0.988297
	EM	1.000016		EM	1.067844		EM	0.979977
	SVDI	0.999691		SVDI	**1.076334**		SVDI	0.979958
	BPCA	0.999633		BPCA	0.996447		BPCA	0.983318
	LLSI	0.999170		LLSI	1.007612		LLSI	0.983508
	EC	1.000539		EC	1.002385		EC	0.944747

The best ranking per data set is stressed in bold

Table 12 Average rankings for Wilson's noise ratio and Mutual information ratio

	Avg. rankings	
	Wilson's Noise ratio	Mutual information
MC	6.98 (8)	8.05 (11)
CMC	**3.79** (1)	**3.60** (1)
KNNI	6.43 (7)	7.69 (8)
WKNNI	6.17 (5)	7.79 (9)
KMI	7.38 (10)	7.60 (6)
FKMI	6.36 (6)	7.62 (7)
SVMI	4.67 (2)	4.90 (4)
EM	8.93 (12)	7.90 (10)
SVDI	8.86 (11)	8.48 (12)
BPCA	7.17 (9)	5.79 (5)
LLSI	5.98 (4)	4.74 (3)
EC	**5.31** (3)	**3.86** (2)

The best methods are stressed in bold

- The EC method is the best method obtained for the approximative models and the third best for the rule induction learning methods. In the latter case, the average ranking of EC is 4.89, very close to the average ranking 4.44 and 4.56 of FKMI and SVMI, respectively.
- The CMC method is the second best method for the lazy learning models and very close to the first one (MC) with an average ranking of 3.63.

Next, we rank all the imputation methods according to the values presented in Tables 10 and 11. In order to do so, we have calculated the average rankings of each imputation method for all the data sets, for both Wilson's noise ratio and the mutual information ratio. The method to compute this average ranking is the same as that presented in Sect. 4.2. In Table 12, we have gathered together these average rankings, as well as their relative position in parentheses.

From the average rankings shown in Table 12, we can observe that the CMC method is the first for both rankings. The EC method is the second for the mutual information ratio and the third one for Wilson's noise ratio. The SVMI method obtains the second lowest ranking for Wilson's noise ratio and the fourth lowest ranking for the MI ratio. The SVMI method is the second best method for the rule induction learning algorithms with average rankings close to EC.

With the analysis performed, we have quantified the noise induced by the imputation methods and how the relationship between each input attribute and the class is maintained. We have discovered that the CMC and EC methods show good behavior for these two measures, and they are two methods that provide good results for an important range of learning methods, as we have previously analyzed. In short, these two approaches introduce less noise and maintain the mutual information better. They can provide us with a first characterization of imputation methods and a first step for providing us with tools for analyzing the imputation method's behavior.

6 Lessons learned

This study is a general comparison of classification methods not previously considered in MV studies, arranged into three different groups, analyzing the best imputation choice by means of non-parametric statistical test. The results obtained agree with the previous studies:

- The imputation methods that fill in the MVs outperform the case deletion (IM method) and the lack of imputation (DNI method).
- There is no universal imputation method that performs best for all classifiers.

From the results seen in Sect. 4.2, the use of the *FKMI* and *EC* imputation methods is the best choice under general assumptions but showing little advantage with respect to the rest of imputation methods analyzed.

According to the results in Sect. 4.6, the particular analysis of the MVs treatment methods conditioned to the classification methods' groups seems necessary. Thus, we can stress the recommended imputation algorithms to be used based on the classification method's type, as in case of the *FKMI* imputation method for the rule induction learning group, the *EC* method for the approximate models and the *MC* method for the lazy learning models. We can confirm the positive effect of the imputation methods and the classifiers' behavior and the presence of more suitable imputation methods for some particular classifier categories than others.

In Sect. 5 we have analyzed the influence of the imputation methods in the data with respect to two measures. These two measures are the *Wilson's noise ratio* and the *average mutual information difference*. The first one quantifies the noise induced by the imputation method in the instances which contain MVs. The second one examines the increment or decrement in the relationship of the isolated input attributes with respect to the class label. We have observed that the CMC and EC methods are the ones which introduce less noise and maintain the mutual information better.

Acknowledgments This work was supported by the Spanish Ministry of Science and Technology under Project TIN2008-06681-C06-01. J. Luengo holds a FPU scholarship from Spanish Ministry of Education and Science.

References

1. Acuna E, Rodriguez C (2004) Classification, clustering and data mining applications. Springer, Berlin, pp 639–648
2. Alcalá-fdez J, Sánchez L, García S, Jesus MJD, Ventura S, Garrell JM, Otero J, Bacardit J, Rivas VM, Fernández JC, Herrera F (2009) Keel: a software tool to assess evolutionary algorithms for data mining problems. Soft Comput 13(3):307–318
3. Asuncion A, Newman D (2007) UCI machine learning repository. http://archive.ics.uci.edu/ml/
4. Atkeson CG, Moore AW, Schaal S (1997) Locally weighted learning. Artif Intell Rev 11:11–73
5. Barnard J, Meng X (1999) Applications of multiple imputation in medical studies: From aids to nhanes. Stat Methods Med Res 8(1):17–36
6. Batista G, Monard M (2003) An analysis of four missing data treatment methods for supervised learning. Appl Artif Intell 17(5):519–533
7. Bezdek J, Kuncheva L (2001) Nearest prototype classifier designs: an experimental study. Int J Intell Syst 16(12):1445–1473
8. Broomhead D, Lowe D (1988) Multivariable functional interpolation and adaptive networks. Complex Syst 11:321–355
9. Clark P, Niblett T (1989) The cn2 induction algorithm. Mach Learn J 3(4):261–283
10. Cohen W (1995) Fast effective rule induction. In: Machine learning: proceedings of the twelfth international conference, pp 1–10
11. Cohen W, Singer Y (1999) A simple and fast and and effective rule learner. In: Proceedings of the sixteenth national conference on artificial intelligence, pp 335–342
12. Cover TM, Thomas JA (1991) Elements of information theory, 2nd edn. Wiley, NY
13. Demšar J (2006) Statistical comparisons of classifiers over multiple data sets. J Mach Learn Res 7:1–30
14. Ding Y, Simonoff JS (2010) An investigation of missing data methods for classification trees applied to binary response data. J Mach Learn Res 11:131–170
15. Domingos P, Pazzani M (1997) On the optimality of the simple bayesian classifier under zero-one loss. Mach Learn 29:103–137

16. Ennett CM, Frize M, Walker CR (2001) Influence of missing values on artificial neural network performance. Stud Health Technol Inform 84:449–453
17. Fan R-E, Chen P-H, Lin C-J (2005) Working set selection using second order information for training support vector machines. J Mach Learn Res 6:1889–1918
18. Farhangfar A, Kurgan LA, Pedrycz W (2007) A novel framework for imputation of missing values in databases. IEEE Trans Syst Man Cybern Part A 37(5):692–709
19. Farhangfar A, Kurgan L, Dy J (2008) Impact of imputation of missing values on classification error for discrete data. Pattern Recognit 41(12):3692–3705
20. Fayyad U, Irani K (1993) Multi-interval discretization of continuous-valued attributes for classification learning. In: Proceedings of 13th international joint conference on uncertainly in artificial intelligence (IJCAI93), pp. 1022–1029
21. Feng H, Guoshun C, Cheng Y, Yang B, Chen Y (2005) A svm regression based approach to filling in missing values. In: Khosla R, Howlett RJ, Jain LC (eds) 'KES (3)', vol 3683 of lecture notes in computer science. Springer, Berlin, pp 581–587
22. Frank E, Witten I (1998) Generating accurate rule sets without global optimization. In: Proceedings of the fifteenth international conference on machine learning, pp 144–151
23. García-Laencina P, Sancho-Gómez J, Figueiras-Vidal A (2009) Pattern classification with missing data: a review. Neural Comput Appl. 9(1):1–12
24. García S, Herrera F (2008) An extension on "statistical comparisons of classifiers over multiple data sets" for all pairwise comparisons. J Mach Learn Res 9:2677–2694
25. Gheyas IA, Smith LS (2010) A neural network-based framework for the reconstruction of incomplete data sets. Neurocomputing In Press, Corrected Proof
26. Grzymala-Busse J, Goodwin L, Grzymala-Busse W, Zheng X (2005) Handling missing attribute values in preterm birth data sets. In: Proceedings of 10th international conference of rough sets and fuzzy sets and data mining and granular computing(RSFDGrC), pp 342–351
27. Grzymala-Busse JW, Hu M (2000) A comparison of several approaches to missing attribute values in data mining. In: Ziarko W, Yao YY (eds) Rough sets and current trends in computing, vol 2005 of lecture notes in computer science, Springer, pp 378–385
28. Hruschka ER Jr., Hruschka ER, Ebecken NF (2007) Bayesian networks for imputation in classification problems. J Intell Inf Syst 29(3):231–252
29. Kim H, Golub GH, Park H (2005) Missing value estimation for dna microarray gene expression data: local least squares imputation. Bioinformatics 21(2):187–198
30. Kwak N, Choi C-H (2002) Input feature selection by mutual information based on parzen window. IEEE Trans Pattern Anal Mach Intell 24(12):1667–1671
31. Kwak N, Choi C-H (2002) Input feature selection for classification problems. IEEE Trans Neural Netw 13(1):143–159
32. Cessie S le, van Houwelingen J (1992) Ridge estimators in logistic regression. Appl Stat 41(1):191–201
33. Li D, Deogun J, Spaulding W, Shuart B (2004) Towards missing data imputation: a study of fuzzy k-means clustering method. In: Proceedings of 4th international conference of rough sets and current trends in computing (RSCTC), pp 573–579
34. Little RJA, Rubin DB (1987) Statistical analysis with missing data, wiley series in probability and statistics, 1st edn. Wiley, New York
35. Luengo J, García S, Herrera F (2010) A study on the use of imputation methods for experimentation with Radial Basis Function Network classifiers handling missing attribute values: the good synergy between RBFNs and EventCovering method. Neural Netw 23(3):406–418
36. Matsubara ET, Prati RC, Batista GEAPA, Monard MC (2008) Missing value imputation using a semi-supervised rank aggregation approach. In: Zaverucha G, da Costa ACPL (eds) 'SBIA', vol 5249 of lecture notes in computer science. Springer, Berlin, pp 217–226
37. McLachlan G (2004) Discriminant analysis and statistical pattern recognition. Wiley, NY
38. Merlin P, Sorjamaa A, Maillet B, Lendasse A (2010) X-SOM and L-SOM: a double classification approach for missing value imputation. Neurocomputing 73(7–9):1103–1108
39. Michalksi R, Mozetic I, Lavrac N (1986) The multipurpose incremental learning system aq15 and its testing application to three medical domains. In: Proceedings of 5th international conference on artificial intelligence (AAAI), pp 1041–1045
40. Moller F (1990) A scaled conjugate gradient algorithm for fast supervised learning. Neural Netw 6: 525–533
41. Nogueira BM, Santos TRA, Zárate LE (2007) Comparison of classifiers efficiency on missing values recovering: application in a marketing database with massive missing data. In: 'CIDM', IEEE, pp 66–72
42. Oba S, aki Sato M, Takemasa I, Monden M, ichi Matsubara K, Ishii S (2003) A bayesian missing value estimation method for gene expression profile data. Bioinformatics 19(16):2088–2096

43. Peng H, Long F, Ding C (2005) Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy. IEEE Trans Pattern Anal Mach Intell 27(8):1226–1238
44. Pham DT, Afify AA (2005) Rules-6: a simple rule induction algorithm for supporting decision making. In: Industrial electronics society, 2005. IECON 2005. 31st annual conference of IEEE, pp 2184–2189
45. Pham DT, Afify AA (2006) Sri: A scalable rule induction algorithm. Proc Inst Mech Eng Part C J Mech Eng Sci 220:537–552
46. Plat J (1991) A resource allocating network for function interpolation. Neural Comput 3(2):213–225
47. Platt JC (1999) Fast training of support vector machines using sequential minimal optimization. In: Advances in kernel methods: support vector learning. MIT Press, Cambridge, pp 185–208
48. Pyle D (1999) Data preparation for data mining. Morgan Kaufmann, Los Altos
49. Qin B, Xia Y, Prabhakar S (2010) Rule induction for uncertain data. Knowl Inf Syst, doi:10.1007/s10115-010-0335-7, pp 1–28 (in press)
50. Quinlan J (1993) C4.5:programs for machine learning. Morgan Kauffman, Los Altos
51. Reddy C, Park J-H (2010) Multi-resolution boosting for classification and regression problems. Knowl Inf Syst, doi:10.1007/s10115-010-0358-0, pp 1–22, (in press)
52. Saar-Tsechansky M, Provost F (2007) Handling missing values when applying classification models. J Learn Res 8:1623–1657
53. Safarinejadian B, Menhaj M, Karrari M (2010) A distributed EM algorithm to estimate the parameters of a finite mixture of components. Knowl Inf Syst 23(3):267–292
54. Schafer JL (1997) Analysis of incomplete multivariate data. Chapman & Hall, London
55. Schneider T (2001) Analysis of incomplete climate data: estimation of mean values and covariance matrices and imputation of missing values. J Clim 14:853–871
56. Song Q, Shepperd M, Chen X, Liu J (2008) Can k-NN imputation improve the performance of C4.5 with small software project data sets? A comparative evaluation. J Syst Softw 81(12):2361–2370
57. Troyanskaya O, Cantor M, Sherlock G, Brown P, Hastie T, Tibshirani R, Botstein D, Altman RB (2001) Missing value estimation methods for dna microarrays. Bioinformatics 17(6):520–525
58. Twala B (2009) An empirical comparison of techniques for handling incomplete data using decision trees. Appl Artif Intell 23:373–405
59. Unnebrink K, Windeler J (n.d.)
60. Wang H, Wang S (2010) Mining incomplete survey data through classification. Knowl Inf Syst 24(2):221–233
61. Wilson D (1972) Asymptotic properties of nearest neighbor rules using edited data. IEEE Trans Syst Man Cybern 2(3):408–421
62. Wong AKC, Chiu DKY (1987) Synthesizing statistical knowledge from incomplete mixed-mode data. IEEE Trans Pattern Anal Mach Intell 9(6):796–805
63. Wu X, Urpani D (1999) Induction by attribute elimination. IEEE Trans Knowl Data Eng 11(5):805–812
64. Zheng Z, Webb GI (2000) Lazy learning of bayesian rules. Mach Learn 41(1):53–84

Author Biographies

Julián Luengo received the M.S. degree in computer science and the PhD degree from the University of Granada, Granada, Spain, in 2006 and 2011, respectively. He is currently holding a Post-Doctoral Research Fellow at the University of Granada. His research interests include machine learning and data mining, data preparation in knowledge discovery and data mining, missing values, data complexity, and fuzzy systems.

Salvador García received the M.Sc. and PhD degrees in computer science from the University of Granada, Granada, Spain, in 2004 and 2008, respectively. He is currently an Assistant Professor in the Department of Computer Science, University of Jaén, Jaén, Spain. He has published more than 25 papers in international journals. As edited activities, he has co-edited two special issues in international journals on different data mining topics. His research interests include data mining, data reduction, data complexity, imbalanced learning, semi-supervised learning, statistical inference, and evolutionary algorithms.

Francisco Herrera received the M.Sc. degree in Mathematics in 1988 and the PhD degree in Mathematics in 1991, both from the University of Granada, Spain. He is currently a Professor in the Department of Computer Science and Artificial Intelligence at the University of Granada. He has published more than 200 papers in international journals. As edited activities, he has co-edited four international books and co-edited twenty special issues in international journals on different soft computing topics. He acts as associated editor of the journals: IEEE Transactions on Fuzzy Systems, Mathware and Soft Computing, Advances in Fuzzy Systems, Advances in Computational Sciences and Technology, and International Journal of Applied Metaheuristic Computing. He currently serves as area editor of the Journal Soft Computing (area of genetic algorithms and genetic fuzzy systems), and he serves as member of the editorial board of the journals: Fuzzy Sets and Systems, Applied Intelligence, Knowledge and Information Systems, Information Fusion, Evolutionary Intelligence, International Journal of Hybrid Intelligent Systems, Memetic Computation, International Journal of Computational Intelligence Research, The Open Cybernetics and Systemics Journal, Recent Patents on Computer Science, Journal of Advanced Research in Fuzzy and Uncertain Systems, and International Journal of Information Technology and Intelligent and Computing. His current research interests include computing with words and decision making, data mining, data preparation, instance selection, fuzzy rule-based systems, genetic fuzzy systems, knowledge extraction based on evolutionary algorithms, memetic algorithms and genetic algorithms.