

# Missing data imputation for fuzzy rule-based classification systems

Julián Luengo · José A. Sáez · Francisco Herrera

Published online: 14 October 2011  
© Springer-Verlag 2011

**Abstract** Fuzzy rule-based classification systems (FRBCSs) are known due to their ability to treat with low quality data and obtain good results in these scenarios. However, their application in problems with missing data are uncommon while in real-life data, information is frequently incomplete 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 FRBCSs considering 14 different approaches to missing attribute values treatment that are presented and analyzed. The analysis involves three different methods, in which we distinguish between Mamdani and TSK models. From the obtained results, the convenience of using imputation methods for FRBCSs with missing values is stated. The analysis suggests that each type behaves differently while the use of determined missing values imputation methods could improve the accuracy obtained for these methods. Thus, the use of particular imputation methods conditioned to the type of FRBCSs is required.

**Keywords** Classification · Missing values · Fuzzy rule-based classification systems · Imputation

## 1 Introduction

Many existing, industrial and research datasets 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 requires a preprocessing stage in which the data are prepared and cleaned (Pyle 1999), in order to be useful to and sufficiently clear for the knowledge extraction process. The simplest way of dealing with missing values is to discard the examples that contain them. However, this method is practical only when the data contains a relatively small number of examples with MVs and when analysis of the complete examples will not lead to serious bias during the inference (Little and Rubin 1987).

Fuzzy rule-based classification systems (FRBCSs) (Ishibuchi et al. 2004; Kuncheva 2000) are widely employed due to their capability to build a linguistic model interpretable to the users with the possibility of mixing different information. They are also well known for being able to deal with imprecise data. However, few analysis have been carried out considering the presence of MVs (Berthold and Huber 1998; Gabriel and Berthold 2005) for FRBCSs and usually the presence of MVs is not usually taken into account and they are usually discarded, maybe inappropriately. Incomplete data in either the training set or test set or in both sets affect the prediction accuracy of learned classifiers (Gheyas and Smith 2010). The seriousness of this problem depends in part on the proportion of missing data. Most FRBCSs cannot work directly with incomplete datasets and due to the high dimensionality of real problems it is possible that no valid (complete) cases would be present in the dataset (García-Laencina et al. 2009).

This inappropriate handling of missing data in the analysis may introduce bias and can result in misleading

---

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

J. A. Sáez  
e-mail: smja@decsai.ugr.es

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

conclusions being drawn from a research study, and can also limit the generalizability of the research findings (Wang and Wang 2010). Three types of problems are usually associated with missing values in data mining (Barnard and Meng 1999): (1) loss of efficiency; (2) complications in handling and analyzing the data; and (3) bias resulting from differences between missing and complete data.

Therefore, the treatment of missing data in data mining is necessary and it can be handled in three different ways normally (Farhangfar et al. 2007):

- The first approach is to discard the examples with missing data in their attributes. Therefore, deleting attributes with elevated levels of missing data are 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 dataset's attributes are not independent from 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 missing data treatment is independent of the learning algorithm used without erasing any example. 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 Neighbour, etc.; to those which analyze the relationships between attributes such as: support vector machines-based, clustering-based, logistic regressions, maximum-likelihood procedures and multiple imputation (Batista and Monard 2003; Farhangfar et al. 2008).

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 (Acuna and Rodriguez 2004; Batista and Monard 2003; Farhangfar et al. 2008; Hruschka Jr. et al. 2007; Li et al. 2004; Luengo et al. 2010).

We want to analyze the effect of the use of a large set of imputation methods on FRBCSs, trying to obtain the best imputation procedure for each one. We consider three representative FRBCSs of different natures which have proven to perform well.

- The fuzzy hybrid-genetic-based machine learning (FH-GBML) method proposed by Ishibuchi et al. (2005) which is a Mamdani-based FRBCS.
- The fuzzy rule learning model proposed by Chi et al. (1996) which is a Mamdani-based FRBCSs as well.
- The positive definite fuzzy classifier (PDFC) proposed by Chen and Wang (2003) which is a Takagi-Sugeno (TSK)-based FRBCS.

In order to perform the analysis, we use a large bunch of datasets, twenty-one in total, with natural MVs. All the datasets 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 missing data treatment, for a total of 14 imputation methods. Therefore, each FRBCS is used over the 14 imputation results. All the imputation and classification algorithms are publicly available in the KEEL software<sup>1</sup> (Alcalá-Fdez et al. 2009). These results are compared using the Wilcoxon Signed Rank test (Demšar 2006; García and Herrera 2008) in order to obtain the best method(s) for each FRBCS. With this information we can extract the best imputation method for each FRBCS, and indicate if there is a common best option depending on the FRBCS type.

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.

The rest of the paper is organized as follows: Sect. 2 introduces the descriptions of the FRBCSs considered and a brief review of the current state of the art in MVs for FRBCSs. In Sect. 3 we present the basis of the application of the imputation methods and the description of the imputation methods we have used. In Sect. 4, the experimental framework, the classification methods and the parameters used for both imputation and classification methods are presented. In Sect. 5, the results obtained are analyzed. In Sect. 6, we use two measures to quantify the influence of the imputation methods in the datasets, both in the instances and in the features. Finally, in Sect. 7 we make some concluding remarks.

## 2 Fuzzy rule-based classification systems

In this section, we describe the basis of the three models that we have used in our study. First, we introduce the

<sup>1</sup> <http://keel.es>.

basic notation that we will use later. Next, we describe the Chi method (Sect. 2.1), the FH-GBML method (Sect. 2.2) and the PDFC method (Sect. 2.3). In Sect. 2.4, we describe the contributions made to the MVs treatment for FRBCSs and we tackle the different situations which apply for the three FRBCSs considered when MVs appear.

Any classification problem consists of  $w$  training patterns  $x_p = (x_{p1}, \dots, x_{pn}), p = 1, 2, \dots, m$  from  $M$  classes where  $x_{pi}$  is the  $i$ th attribute value ( $i = 1, 2, \dots, n$ ) of the  $p$ -th training pattern.

In this work, we use fuzzy rules in the following form:

$$\text{Rule } R_j : \text{If } x_1 \text{ is } A_j^1 \text{ and } \dots \text{ and } x_n \text{ is } A_j^n \text{ then Class} = C_j \text{ with } RW_j \tag{1}$$

where  $R_j$  is the label of the  $j$ th rule,  $x = (x_1, \dots, x_n)$  is an  $n$ -dimensional pattern vector,  $A_j^i$  is an antecedent fuzzy set,  $C_j$  is a class label or a numeric value, and  $RW_j$  is the rule weight. We always use triangular membership functions as antecedent fuzzy sets.

### 2.1 Chi et al. approach

This FRBCSs design method (Chi et al. 1996) is an extension of the well-known Wang and Mendel method (1992) for classification problems. To generate the fuzzy rule base (RB), it determines the relationship between the variables of the problem and establishes an association between the space of the features and the space of the classes by means of the following steps:

Step 1: *Establishment of the linguistic partitions.* Once the domain of variation of each feature  $A_i$  is determined, the fuzzy partitions are computed.

Step 2: *Generation of a fuzzy rule for each example*  $x_p = (x_{p1}, \dots, x_{pn}, C_p)$ . To do this it is necessary:

Step 2.1: To compute the matching degree  $\mu(x_p)$  of the example to the different fuzzy regions using a conjunction operator (usually modeled with a minimum or product T-norm).

Step 2.2: To assign the example  $x_p$  to the fuzzy region with the greatest membership degree.

Step 2.3: To generate a rule for the example, whose antecedent is determined by the selected fuzzy region and whose consequent is the label of class of the example.

Step 2.4: To compute the rule weight.

We must remark that rules with the same antecedent can be generated during the learning process. If they have the same class in the consequent we just remove one of the

duplicated rules, but if they have a different class only the rule with the highest weight is kept in the RB.

### 2.2 Fuzzy hybrid-genetic-based machine learning rule generation algorithm

The basis of the algorithm described here (Ishibuchi et al. 2005) consists of a Pittsburgh approach where each rule set is handled as an individual. It also contains a genetic cooperative-competitive learning (GCCL) approach (an individual represents a unique rule), which is used as a kind of heuristic mutation for partially modifying each rule set, because of its high search ability to efficiently find good fuzzy rules.

The system defines 14 possible linguistic terms for each attribute, as shown in Fig. 1, which correspond to Ruspini's strong fuzzy partitions with two, three, four, and five uniformly distributed triangular-shaped membership functions. Furthermore, the system also uses "don't care" as an additional linguistic term, which indicates that the variable matches any input value with maximum matching degree.

The main steps of this algorithm are described below:

Step 1: Generate  $N_{pop}$  rule sets with  $N_{rule}$  fuzzy rules.

Step 2: Calculate the fitness value of each rule set in the current population.

Step 3: Generate  $(N_{pop} - 1)$  rule sets by selection, crossover and mutation in the same manner as the Pittsburgh-style algorithm. Apply a single iteration of the GCCL-style algorithm (i.e., the rule generation and the replacement) to each of the generated rule sets with a pre-specified probability.

Step 4: Add the best rule set in the current population to the newly generated  $(N_{pop} - 1)$  rule sets to form the next population.

Step 5: Return to Step 2 if the pre-specified stopping condition is not satisfied.

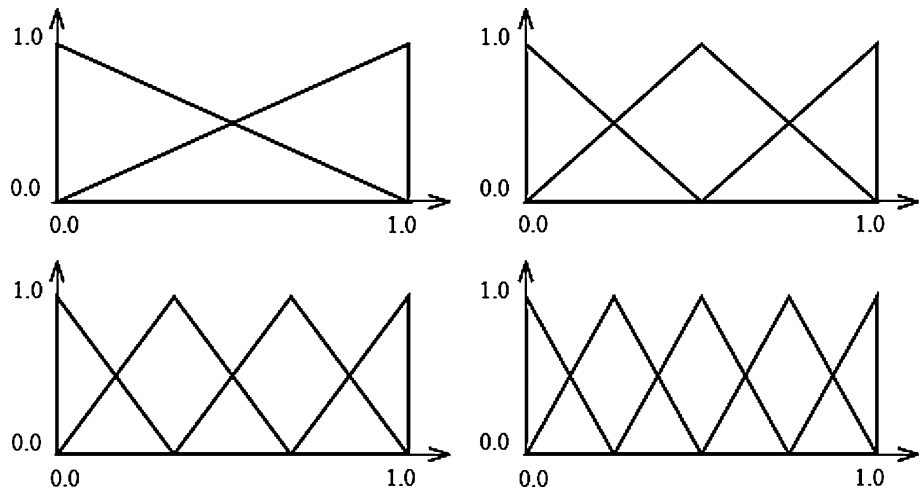
Next, we will describe every process of the algorithm:

- Initialization:  $N_{rule}$  training patterns are randomly selected. Then, a fuzzy rule from each of the selected training patterns is generated by choosing probabilistically [as shown in (2)] an antecedent fuzzy set from the 14 candidates  $B_k (k = 1, 2, \dots, 14)$  (see Fig. 1) for each attribute. Then each antecedent fuzzy set of the generated fuzzy rule is replaced with *don't care* using a pre-specified probability  $P_{don't\ care}$ .

$$P_{don't\ care}(B_k) = \frac{\mu_{B_k}(x_{pi})}{\sum_{j=1}^{14} \mu_{B_j}(x_{pi})} \tag{2}$$

- Fitness computation: the fitness value of each rule set  $S_i$  in the current population is calculated as the number of

**Fig. 1** Four fuzzy partitions for each attribute membership function



correctly classified training patterns by  $S_i$ . For the GCCL approach the computation follows the same scheme.

- Selection: it is based on binary tournament.
- Crossover: the substring-wise and bit-wise uniform crossover are applied in the Pittsburgh part. In the case of the GCCL part only the bit-wise uniform crossover is considered.
- Mutation: each fuzzy partition of the individuals is randomly replaced with a different fuzzy partition using a pre-specified mutation probability for both approaches.

### 2.3 Positive definite function classifier

The PDFC learning method (Chen and Wang 2003) uses a support vector machine (SVM) approach to build up the model. PDFC considers a fuzzy model with  $m + 1$  fuzzy rules of the form given in Eq. (1) where  $A_j^k$  is a fuzzy set with membership function  $a_j^k : \mathbb{R} \rightarrow [0, 1]$ ,  $RW_j = 1$  and  $C_j = b_j \in \mathbb{R}$ . Therefore, PDFC is a FRBCS with constant THEN-parts. If we choose product as the fuzzy conjunction operator, addition for fuzzy rule aggregation and center of area defuzzification, then the model becomes a special form of the Takagi-Sugeno fuzzy model.

PDFC considers the use of membership functions generated from a reference function  $a^k$  through location transformation (Dubois and Prade 1978). In Chen and Wang (2003) well-known types of reference functions can be found, like the symmetric triangle and the gaussian function. As a consequence of the presented formulation,

$$K(x_p, z_j) = \prod_{k=1}^n a^k(x_p^k - z_j^k) \tag{3}$$

is a Mercer Kernel (Cristianini and Shawe-Taylor 2000), if it has nonnegative Fourier transform. Thus, the decision rule of a binary fuzzy classifier is

$$f(x_p) = \text{sign} \left( b_0 + \sum_{j=1}^m b_j \prod_{k=1}^n a_j^k(x_p^k) \right). \tag{4}$$

So the remaining question is how to find a set of fuzzy rules ( $\{z_1, \dots, z_m\}$  and  $\{b_0, \dots, b_m\}$ ). It is well known that the SVM algorithm finds a separating hyperplane with good generalization by reducing the empirical risk and, at the same time, controlling the hyperplane margin (Vapnik 1998). Thus, we can use the SVM algorithm to find an optimal hyperplane in  $\mathbb{F}$ . Once we get such a hyperplane, fuzzy rules can easily be extracted. The whole procedure is described next:

Step 1: Construct a Mercer kernel,  $K$ , from the given positive-definite reference functions according to (3).

Step 2: Construct an SVM to get a decision rule of the form

$$f(x) = \text{sign} \left( \sum_{i \in S} y_i \alpha_i K(x, x_i) + b \right),$$

with  $S$  as the index set of the support vectors:

Step 2.1: Assign some positive number to the cost  $C$ , and solve the quadratic program defined by the proper SVM to get the Lagrange multipliers  $\alpha_i$ .

Step 2.2: Find  $b$  [details can be found in, for example, (Platt 1999)].

Step 3: Extract fuzzy rules from the decision rule of the SVM:

Step 3.1:  $b_0$  is the constant parameter of the hyperplane, that is  $b_0 \leftarrow b$ .

Step 3.2: For each support vector create a fuzzy rule where: we center the reference functions on the support vector  $z_j \leftarrow x_i$  and we assign the rule consequent  $b_j \leftarrow y_i \alpha_i$ .

## 2.4 Missing values treatment for fuzzy rule-based classification systems

Traditionally, the presence of MVs in the data has not been considered when building up the FRBCS model. Although the FRBCS are capable of managing imperfect data, their abilities have not been explicitly checked in this case. The only precedent in the literature of FRBCSs learning in the case of MVs is a technique proposed to tolerate MVs in the training of a FRBCS by Berthold and Huber (1998). This procedure was initially intended to estimate the best approximation to the MV based on the core region of the fuzzy label associated to the missing attribute.

This initial work was further developed applying the initial technique to a particular fuzzy rule induction algorithm in (Gabriel and Berthold 2005). The main idea was to avoid the use of the missing attribute in the rule operations when covering new examples or specializing the rules. This is a simple and easy to implement idea, but its extension is limited to few fuzzy rule induction algorithms, like FH-GBML.

As we can appreciate from the mentioned studies there is a lack of research in this area. There are many different approaches to the treatment of MVs, which use many different methods (to classify and to impute MVs), but they have not been considered with FRBCSs. Therefore, in spite of the variety of studies presented, the necessity of analyze the use of imputation methods for FRBCSs is demonstrated.

Only one of the presented classification methods in the previous section has its own MVs treatment. We have applied the procedure indicated in (Gabriel and Berthold 2005) using the “don’t care” label, but this extension is not easy to apply to Chi et al. and PDFC algorithms due to their different nature. For this reason PDFC and Chi et al. FRBCSs are not able to deal with MVs. Thus we set the training and test accuracy to zero in the presence of MVs, as the methods cannot build a model or compute a distance to the instance.

## 3 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. 3.1 we indicate the fundamental aspects in the MVs treatment based on the MV introduction mechanism. In Sect. 3.2, 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 url:<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.

### 3.1 Missing values introduction mechanisms

It is important to categorize the mechanisms which lead to the introduction of MVs (Little and Rubin 1987). The assumptions we make about the missingness mechanism and the missing data pattern of missing values can affect which imputation method could be applied, if any. As Little and Rubin (1987) stated, there are three different mechanisms for missing data 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 missing data.
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 missing data.
3. Not missing at random (NMAR), when the distribution of an example having a missing value for an attribute depends on the missing values.

In the 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 missing data can be predicted using the complete data (Little and Rubin 1987). These two mechanisms are assumed by the imputation methods so far. As Farhangfar et al. (2008) and Matsubara et al. (2008) stated, 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 missing data should be applied only if the missing data 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 missing values to a new value (encode them into a new numerical value), but such a simplistic method was shown to lead to serious inference problems (Schafer 1997). On the other hand, if a significant number of examples contain missing values for a relatively small number of attributes, it may be beneficial to perform imputation (filling-in) of the missing values. In order to do so, the assumption of MAR randomness is needed, as Little and Rubin (1987) observed in their analysis.

In our case we will use single imputation methods, due to the time complexity 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 missing data prior to their application.

### 3.2 Description of the imputation methods

In this subsection, we briefly describe the imputation methods that we have used.

- Do not impute (DNI): as its name indicates, all the missing data remains unreplaced, so the FRBCSs 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 datasets. As a guideline, in Grzymala-Busse and Hu (2000) 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 dataset.
- Global most common attribute value for symbolic attributes, and global average value for numerical attributes (MC) (Grzymala-Busse et al. 2005): 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) (Grzymala-Busse et al. 2005): as stated in *MC*, the MV is replaced by the most repeated one if nominal or the mean value if numerical, but considering only the instances with the same class as the reference instance.
- Imputation with K-nearest neighbor (KNNI) (Batista and Monard 2003): 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 the most commonly used in the literature.
- Weighted imputation with K-nearest neighbor (WKNNI) (Troyanskaya et al. 2001): 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) (Li et al. 2004): given a set of objects, the overall objective of clustering is to divide the dataset 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 missing data, in a similar way to KNNI.
- Imputation with fuzzy K-means clustering (FKMI) (Acuna and Rodriguez 2004; Li et al. 2004): 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's only take 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) (Feng et al. 2005) is an SVM regression-based algorithm to fill in missing data, 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) (Wong and Chiu 1987): based on the work of Wong and Chiu (1987), 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) (Schneider 2001): missing values 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 missing values, the regression parameters of the variables with missing values among the variables with available values are computed from the estimates of the mean and of the covariance matrix. Second, the missing values 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 dataset and the covariance matrix as the sum of the sample covariance matrix of the completed dataset 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) (Troyanskaya et al. 2001): 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 dataset. 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, to obtain an estimation of the MV itself.
- Bayesian principal component analysis (BPCA) (Oba et al. 2003): this method is an estimation method for missing values, 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 expectation-maximization (EM)-like repetitive algorithm.

- Local least squares imputation (LLSI) (Kim et al. 2005): with this method, a target instance that has missing values are 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.

## 4 Experimental framework

When analyzing imputation methods, a wide range of set ups can be observed. The datasets 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. 4.1.

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 classification methods. Therefore, they must be indicated in order to be able to reproduce any results obtained. In Sect. 4.2 the parameter configurations used by all the methods considered in this study are presented.

### 4.1 Datasets description

The experimentation has been carried out using 21 benchmark datasets from the KEEL-Dataset repository.<sup>2</sup> Each dataset 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 cannot know anything about the randomness of MVs in the datasets, so we assume they are distributed in an *MAR* way, so the application of the imputation methods is feasible. In our study, we want to deal with the original

<sup>2</sup> <http://sci2s.ugr.es/keel/datasets.php>.

**Table 1** Datasets used

Dataset	Acronym	# Instances	# Attributes	# Classes	% MV	% Instance 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

MVs and therefore to obtain the real accuracy values of each dataset with our imputation methods. In addition to this, we use all kinds of datasets, which include nominal datasets, numeric datasets and mixed-mode datasets.

In order to carry out the experimentation, we have used a tenfold 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.

All these datasets 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 dataset 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.

#### 4.2 Parameter configuration

In Table 2 we show the parameters used by each imputation method described in Sect. 3.2, in cases where the method needs a parameter. The values chosen are those as recommended by their respective authors. Please refer to their respective papers for further descriptions of the parameters' meaning.

In Table 3, the parameters used by the different FRBCSs are presented. All these parameters are the recommended ones that have been extracted from the respective publications of the methods. Please refer to the associated publications and the KEEL platform to obtain further details about the meaning of the different parameters.

### 5 Analysis of the imputation methods for fuzzy rule-based classification systems

In this section, we analyze the imputation results obtained for the FRBCSs and study the best imputation choices in each case. We first show the test accuracy results for the three FRBCSs using the 14 imputation methods in Sect. 5.1 and indicate the best approaches using these initial criteria.



**Table 2** Imputation methods parameters

Method	Parameters
SVM	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

In order to establish a more robust and significant comparison we have used the Wilcoxon Signed Rank test in Sect. 5.2, to support our analysis with a statistical test that provides us with statistical evidence of the good behavior of any imputation approach for the FRBCSs.

### 5.1 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 for each FRBCS. Following the indications given in the previous subsection, in Table 4 we depict the average test accuracy for the three FRBCSs for each imputation method and dataset. The best imputation method in each case is stressed in bold face. We include a final column with the average accuracy across all datasets for each imputation method.

Attending to the average test accuracy obtained, the best imputation methods are:

- DNI for FH-GBML: the use of the “don’t care” option when MVs appear obtains good results in comparison

**Table 3** Parameters used by the FRBCSs ( $p$  is the number of attributes in the dataset)

FH-GBML
Number of fuzzy rules: $5 \times p$ rules
Number of rule sets ( $N_{pop}$ ): 200 rule sets
Crossover probability: 0.9
Mutation probability: $1/p$
Number of replaced rules: All rules except the best-one (Pittsburgh-part, elitist approach), number of rules/5 (Michigan-part)
Total number of generations: 1,000 generations
Don’t care probability: 0.5
Probability of the application of the Michigan iteration: 0.5
PDFC
$C = 100$
$d = 0.25$
Positive definite function type: symmetric triangle
Chi
Number of labels: 3
T-norm: product
Rule weight: penalized certainty factor
Fuzzy reasoning method: winning rule

with the rest of imputation methods. This is specially appreciable in the case of HOC and AUD datasets. The CMC method is also close in the final average, while it presents less difference with the best method when it is not the best one unlike DNI.

- BPCA for Chi et al.: although BPCA presents an irregular behavior, its superior performance in DER, HOV, BAN, LUN and HEP datasets allows it to obtain a better average. Again, CMC is the second best method and its behavior is more similar to the rest of the methods with less variations.
- EC for PDFC: the results for PDFC are less irregular, and EC is consistently better in the majority of them. In contraposition with FH-GBML and Chi et al., in this case the best imputation method obtains a clear difference with the second best, SVM in this case.

From these results, an initial recommendation of the best imputation procedures for each FRBCS can be made. However, the high variations in the results discourages to use the accuracy as the criterion to select them, specially for FH-GBML and Chi et al. methods. Therefore, a more robust procedure must be used in the comparisons in order to obtain the best imputation method for each FRBCS. This is discussed in the next subsection.

**Table 4** Test accuracy results for FH-GBML, Chi et al. and PDFC

FRBCS	Imputation method	CLE	WIS	CRX	BRE	AUT	PRT	DER	HOV	WAT	SPO	BAN	HOC	AUD	LUN	HEP	MUS	POS	ECH	SOY	MAM	OZO	Average	
FH-GBML	IM	52.21	<b>96.92</b>	85.25	70.79	<b>50.31</b>	21.61	29.47	91.79	<b>67.70</b>	16.38	65.59	0.00	20.00	0.00	81.82	<b>85.81</b>	62.36	52.39	18.48	83.18	84.91	54.14	
	EC	52.81	94.57	85.21	74.10	33.74	28.57	26.58	93.77	28.35	19.29	<b>67.41</b>	76.64	36.34	0.00	75.50	80.09	64.44	45.44	17.62	82.63	5.96	51.86	
	KNNI		49.82	95.86	84.19	69.22	41.72	21.43	<b>33.96</b>	93.53	62.55	16.96	57.78	49.89	18.68	0.00	74.92	81.43	<b>68.89</b>	59.89	12.33	83.67	86.94	55.41
		WKNNI	53.11	96.14	83.76	71.08	43.73	21.60	21.40	92.63	64.65	8.93	58.15	45.78	11.11	0.00	78.08	81.30	65.56	59.07	15.62	82.73	83.35	54.18
	KMI	53.78	96.29	84.48	72.75	45.91	23.78	21.99	92.85	66.54	6.61	55.93	71.43	20.59	0.00	77.38	80.70	61.11	53.63	19.48	83.46	84.37	55.86	
	FKMI	52.75	96.71	84.63	70.94	41.60	20.99	22.79	92.61	60.85	14.11	62.78	76.34	19.86	0.00	78.75	81.65	63.33	52.97	<b>26.39</b>	83.67	81.49	56.44	
	SVMi	54.48	96.43	84.48	75.51	42.58	25.00	31.28	93.77	64.82	14.82	57.78	45.29	20.47	0.00	81.83	80.11	63.33	52.20	17.85	82.94	86.74	55.80	
	EM	51.83	96.14	84.91	73.11	45.31	8.43	27.65	86.87	54.96	14.46	53.15	75.73	0.00	0.00	77.46	83.15	65.56	56.15	10.42	82.62	76.95	53.57	
	SVDI	52.82	96.42	83.47	73.07	45.89	14.76	27.91	89.17	57.82	13.04	50.19	77.45	0.00	0.00	<b>83.17</b>	78.06	66.67	50.60	14.74	81.69	74.50	53.88	
	BPCA	48.49	43.78	47.31	<b>78.03</b>	49.37	<b>34.26</b>	8.24	49.97	46.75	2.50	52.78	56.28	12.77	<b>25.00</b>	78.08	43.00	51.11	36.37	0.00	51.19	80.67	42.66	
	LLSI	<b>56.08</b>	96.57	83.76	72.71	44.71	24.15	24.92	<b>94.93</b>	61.61	14.46	55.74	63.83	29.58	0.00	73.33	83.68	63.33	57.69	14.35	83.15	83.39	56.28	
	MC	53.45	96.43	84.34	72.71	42.48	25.43	30.40	91.46	64.83	14.64	59.81	76.30	21.76	0.00	78.04	81.43	62.22	50.00	15.66	83.46	84.10	56.62	
	CMC	53.77	96.43	<b>85.65</b>	71.03	46.11	25.36	31.53	94.01	64.28	16.96	55.56	70.17	14.64	0.00	78.63	82.62	63.33	<b>61.59</b>	15.04	<b>83.67</b>	<b>86.94</b>	57.02	
	DNI	54.45	96.14	84.77	73.48	42.21	23.36	26.00	94.70	55.92	<b>21.79</b>	<b>62.59</b>	<b>84.77</b>	<b>42.02</b>	0.00	74.88	75.70	63.33	52.20	24.04	81.90	65.03	<b>57.11</b>	
	Chi et al.	IM	39.69	91.51	<b>81.77</b>	66.88	61.99	9.97	5.07	42.25	64.42	<b>14.95</b>	35.69	0.00	<b>20.00</b>	0.00	18.70	99.47	23.19	<b>61.92</b>	10.20	78.89	91.46	43.71
		EC	<b>44.17</b>	75.11	50.09	65.09	48.80	10.27	4.93	46.55	1.90	10.54	37.22	6.57	5.38	0.00	23.25	99.61	22.22	49.18	9.11	78.98	2.92	32.95
KNNI		39.89	91.42	80.71	65.45	59.51	11.38	4.93	41.93	64.66	11.79	38.52	1.35	0.91	0.00	30.96	93.61	22.22	58.96	9.43	78.67	92.70	42.81	
WKNNI		39.89	91.42	80.71	65.45	59.51	11.38	4.93	41.93	64.66	11.79	38.52	1.35	0.91	0.00	30.96	93.61	22.22	58.96	9.43	78.67	92.78	42.81	
KMI		39.89	91.42	81.14	65.45	59.55	12.70	4.93	41.49	65.03	11.79	29.44	4.06	5.34	0.00	22.29	<b>99.74</b>	22.22	54.56	10.40	78.25	92.26	42.47	
FKMI		39.89	91.28	81.14	65.45	59.55	12.38	4.93	42.17	64.07	11.79	37.22	5.17	0.91	0.00	30.96	<b>99.74</b>	22.22	56.04	<b>11.70</b>	78.04	91.91	43.17	
SVMi		40.23	<b>91.56</b>	80.85	65.45	58.93	9.34	4.93	41.93	64.65	11.79	38.52	1.62	5.45	0.00	30.29	<b>99.74</b>	22.22	60.38	10.42	80.85	92.78	43.43	
EM		39.88	91.13	80.71	66.12	68.82	4.93	44.47	62.00	10.54	29.63	29.37	0.91	0.00	25.13	99.29	22.22	58.41	8.78	79.61	91.95	43.75		
SVDI		40.88	91.28	80.56	66.12	67.36	5.86	4.93	45.62	58.96	10.54	27.78	26.92	0.91	0.00	24.46	96.55	22.22	56.92	8.78	78.77	91.59	43.19	
BPCA		39.60	44.78	49.49	<b>78.42</b>	<b>82.21</b>	<b>67.03</b>	<b>67.03</b>	<b>19.50</b>	<b>50.20</b>	50.19	3.75	<b>56.11</b>	<b>54.37</b>	18.12	<b>28.33</b>	<b>73.54</b>	50.84	<b>42.22</b>	36.37	3.26	51.92	87.41	<b>47.03</b>
LLSI		39.89	91.28	80.85	65.09	54.76	8.29	4.93	47.45	59.71	11.96	32.96	2.17	12.39	0.00	31.58	97.38	22.22	55.16	8.78	79.61	92.34	42.80	
MC		39.89	91.28	81.00	65.45	57.90	12.38	4.93	42.17	64.27	11.79	38.70	2.47	0.91	0.00	31.58	<b>99.74</b>	22.22	59.07	<b>11.70</b>	78.04	91.99	43.21	
CMC		39.89	91.42	80.71	65.45	60.40	10.74	4.93	47.93	<b>65.79</b>	11.79	37.96	11.98	0.91	0.00	31.58	<b>99.74</b>	22.22	59.73	10.41	<b>81.06</b>	<b>92.82</b>	44.16	
DNI		0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
PDFC		IM	52.54	<b>96.48</b>	83.40	73.65	80.96	42.46	88.84	94.92	74.39	70.05	75.39	40.00	20.00	36.67	<b>83.25</b>	<b>100.00</b>	52.50	56.54	91.26	83.13	92.65	70.91
		EC	<b>58.43</b>	96.28	83.03	72.36	85.61	42.70	88.81	94.47	73.78	<b>72.68</b>	<b>79.07</b>	82.62	75.36	<b>40.00</b>	82.58	<b>100.00</b>	55.56	<b>63.57</b>	91.24	82.62	93.49	<b>76.87</b>
	KNNI	53.10	95.57	<b>83.61</b>	71.63	76.71	43.38	88.81	94.23	76.65	<b>72.68</b>	<b>77.22</b>	80.99	75.73	<b>40.00</b>	77.50	<b>100.00</b>	54.44	52.69	89.28	82.94	93.77	75.28	
	WKNNI	53.10	95.57	<b>83.61</b>	71.63	76.71	43.02	88.81	94.23	77.03	<b>72.68</b>	<b>77.22</b>	81.26	75.73	<b>40.00</b>	77.50	<b>100.00</b>	53.33	51.92	89.60	82.94	93.77	75.22	
	KMI	52.12	95.71	82.60	69.88	76.23	43.73	88.81	94.69	76.27	<b>72.68</b>	76.85	83.43	<b>76.66</b>	<b>40.00</b>	79.46	<b>100.00</b>	55.56	53.24	91.24	82.52	94.16	75.52	
	FKMI	52.12	95.57	82.74	69.88	76.71	43.40	88.81	94.93	75.90	<b>72.68</b>	77.78	83.45	76.23	<b>40.00</b>	80.04	<b>100.00</b>	54.44	57.09	91.24	82.94	93.80	75.70	
	SVMi	53.12	95.42	83.46	69.88	77.64	41.83	88.81	93.78	<b>78.36</b>	<b>72.68</b>	78.89	80.98	76.23	<b>40.00</b>	82.58	<b>100.00</b>	<b>56.67</b>	54.84	90.91	82.73	94.04	75.85	
	EM	53.43	95.57	83.17	71.26	83.05	31.04	<b>89.08</b>	90.12	70.01	<b>72.68</b>	74.63	77.65	22.61	<b>40.00</b>	81.38	<b>100.00</b>	54.44	58.52	88.62	83.04	93.25	72.07	
	SVDI	53.13	95.71	82.74	72.66	83.52	32.33	<b>89.08</b>	90.57	70.58	<b>72.68</b>	74.63	77.93	24.86	<b>40.00</b>	79.46	<b>100.00</b>	54.44	55.44	88.62	82.31	93.37	72.10	
	BPCA	38.94	43.92	53.85	<b>93.10</b>	<b>90.58</b>	<b>69.78</b>	19.50	50.20	50.37	3.75	57.22	54.37	18.12	28.33	73.54	50.84	43.33	36.37	3.26	50.36	85.00	48.32	
	LLSI	52.43	95.57	83.18	69.53	77.68	38.38	88.81	94.70	73.80	<b>72.68</b>	78.33	82.34	72.71	<b>40.00</b>	78.71	<b>100.00</b>	53.33	53.96	86.69	82.83	93.73	74.73	
	MC	53.11	95.71	82.75	69.88	75.84	43.46	88.81	94.93	75.69	<b>72.68</b>	77.22	82.91	76.23	<b>40.00</b>	77.50	<b>100.00</b>	55.56	55.55	91.24	82.94	93.37	75.49	
	CMC	52.10	95.42	83.03	69.88	76.71	41.71	88.81	<b>95.16</b>	77.80	<b>72.68</b>	73.70	<b>85.89</b>	75.36	<b>40.00</b>	80.71	<b>100.00</b>	54.44	54.07	<b>91.88</b>	<b>83.77</b>	<b>94.20</b>	75.59	
	DNI	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

**Table 5** Average ranks for FH-GBML

	IM	EC	KNNI	WKNNI	KMI	FKMI	SVMi	EM	SVDI	BPCA	LLSI	MC	CMC	DNI	Ties + wins	Wins	Ranking
IM		D	D	W	D	D	D	W	W	W	D	D	D	D	<b>13</b>	<b>4</b>	<b>3</b>
EC	D		D	D	D	D	D	D	D	W	D	D	D	L	<b>12</b>	<b>1</b>	<b>9.5</b>
KNNI	D	D		D	D	D	D	W	D	W	D	D	D	D	<b>13</b>	<b>2</b>	<b>6</b>
WKNNI	L	D	D		D	D	L	D	D	W	D	L	L	D	<b>9</b>	<b>1</b>	<b>11</b>
KMI	D	D	D	D		D	D	D	D	W	D	D	D	D	<b>13</b>	<b>1</b>	<b>8</b>
FKMI	D	D	D	D	D		D	D	W	W	D	D	D	D	<b>13</b>	<b>2</b>	<b>6</b>
SVMi	D	D	D	W	D	D		W	W	W	D	D	D	D	<b>13</b>	<b>4</b>	<b>3</b>
EM	L	D	L	D	D	D	L		D	W	D	L	L	D	<b>8</b>	<b>1</b>	<b>12.5</b>
SVDI	L	D	D	D	D	L	L	D		W	D	L	L	D	<b>8</b>	<b>1</b>	<b>12.5</b>
BPCA	L	L	L	L	L	L	L	L	L		L	L	L	L	<b>0</b>	<b>0</b>	<b>14</b>
LLSI	D	D	D	D	D	D	D	D	D	W		D	L	D	<b>12</b>	<b>1</b>	<b>9.5</b>
MC	D	D	D	W	D	D	D	W	W	W	D		D	D	<b>13</b>	<b>4</b>	<b>3</b>
CMC	D	D	D	W	D	D	D	W	W	W	W	D		D	<b>13</b>	<b>5</b>	<b>1</b>
DNI	D	W	D	D	D	D	D	D	D	W	D	D	D		<b>13</b>	<b>2</b>	<b>6</b>

5.2 Statistical analysis

In order to appropriately analyze the imputation and classification methods, we apply the Wilcoxon Signed rank test comparing the imputation methods for each FRBCS separately. With the results of the test, we create one table per FRBCS in which we provide an average ranking for each imputation method indicating the best ones. The content of the tables and its interpretation is 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  $\alpha$  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  $\alpha$  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 FRBCS. 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 for all of them.

In order to compare the imputation methods for the FRBCS considered in each we have added one more final column with the mean ranking for each imputation method across all the datasets, that is, the mean of every row. By doing so, we can obtain a new rank (final column RANKING), in which we propose a new ordering for the imputation methods for a given FRBCS, using the values of the column “Avg.” to sort the imputation methods.

Table 5 depicts the results for FH-GBML. The best imputation method is CMC, while DNI is the sixth best. That means that although DNI is capable of obtaining very good results in few datasets, it is a very irregular MV treatment strategy. CMC is capable of obtaining good results in every dataset and even being the best on some of them. SVMi and MC imputation methods are also good alternatives for FH-GBML. Case deletion (IM) is an affordable option due to the good generalization abilities of FH-GBML, but it obtains a lower mean accuracy than SVMi and MC.

Table 6 summarize the results for Chi et al. Again we find that CMC is the best imputation choice, while BPCA was the preferred one considering only the accuracy results. The behavior of BPCA is even more irregular for Chi et al. than DNI for FH-GBML. CMC is better than most of the imputation strategies, with 9 wins, making it a clear imputation choice for this FRBCS.

Table 7 shows the results for PDFC. In this case, the ranking using the Wilcoxon statistical comparison is in concordance with the results obtained using the test accuracy: EC is the best imputation method for PDFC with 10 wins out of 14 methods. We can conclude that EC is the best imputation method for PDFC.

For the Wilcoxon tables with their rankings we have built Table 8 with the best three methods of each FRBCS. We have stressed in bold those rankings equal to or below three. An important outcome of the results is that both FH-GBML and Chi et al. FRBCSs share the same best

**Table 6** Average ranks for Chi et al.

	IM	EC	KNNI	WKNNI	KMI	FKMI	SVMi	EM	SVDI	BPCA	LLSI	MC	CMC	DNI	Ties + wins	Wins	Ranking
IM		W	D	D	D	D	D	D	D	D	W	D	D	W	13	3	3
EC	L		L	L	L	L	L	L	L	L	L	L	L	W	1	1	13
KNNI	D	W		D	D	D	L	D	D	D	D	D	L	W	11	2	10.5
WKNNI	D	W	D		D	D	L	D	D	D	D	D	L	W	11	2	10.5
KMI	D	W	D	D		D	D	D	D	D	D	D	L	W	12	2	7.5
FKMI	D	W	D	D	D		D	D	D	D	D	D	L	W	12	2	7.5
SVMi	D	W	W	W	D	D		D	D	D	D	D	D	W	13	4	2
EM	D	W	D	D	D	D	D		W	D	D	D	L	W	12	3	4
SVDI	D	W	D	D	D	D	D	L		D	D	D	L	W	11	2	10.5
BPCA	D	W	D	D	D	D	D	D	D		D	D	D	W	13	2	5.5
LLSI	L	W	D	D	D	D	D	D	D	D		D	L	W	11	2	10.5
MC	D	W	D	D	D	D	D	D	D	D	D		D	W	13	2	5.5
CMC	D	W	W	W	W	W	D	W	W	D	W	D		W	13	9	1
DNI	L	L	L	L	L	L	L	L	L	L	L	L	I		0	0	14

**Table 7** Average ranks for PDFC

	IM	EC	KNNI	WKNNI	KMI	FKMI	SVMi	EM	SVDI	BPCA	LLSI	MC	CMC	DNI	Ties + wins	Wins	Ranking
IM		L	D	D	D	D	D	D	D	W	D	D	D	W	12	2	7.5
EC	W		W	W	D	W	D	W	W	W	W	D	W	W	13	10	1
KNNI	D	L		D	D	D	L	D	D	W	D	D	D	W	11	2	9.5
WKNNI	D	L	D		D	D	D	D	D	W	D	D	D	W	12	2	7.5
KMI	D	D	D	D		D	D	D	W	W	D	D	D	W	13	3	6
FKMI	D	L	D	D	D		D	D	W	W	W	D	D	W	12	4	4.5
SVMi	D	D	W	D	D	D		W	W	W	W	D	D	W	13	6	2
EM	D	L	D	D	D	D	L		D	W	D	D	D	W	11	2	9.5
SVDI	D	L	D	D	L	L	L	D		W	D	L	L	W	7	2	12
BPCA	L	L	L	L	L	L	L	L	L		L	L	L	W	1	1	13
LLSI	D	L	D	D	D	L	L	D	D	W		L	L	W	8	2	11
MC	D	D	D	D	D	D	D	D	W	W	W		D	W	13	4	3
CMC	D	L	D	D	D	D	D	D	W	W	W	D		W	12	4	4.5
DNI	L	L	L	L	L	L	L	L	L	L	L	L	L		0	0	14

imputation method, while PDFC has a different choice. We must stress that FH-GBML and Chi et al. are Mamdani-based FRBCSs, while PDFC is a special form of TSK model. Therefore, the kind of FRBCS considered appears to have influence on the best imputation strategy when considering other FRBCSs than those analyzed in this work.

As a final remark, we can state that:

- The imputation methods which fill in the MVs outperform the case deletion (IM method) and the lack of imputation (DNI method). Only in the case of the IM imputation method obtain a relatively low rank (2nd and 3th place) but these results can be altered when new examples are presented to the model learned with less data. This fact indicates that the imputation methods usually outperform the non-imputation strategies.
- There is no universal imputation method which performs best for all type of FRBCS.

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.

### 6 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 FRBCS’s accuracy. However, it would be interesting

**Table 8** Best imputation methods for FRBCS

	FH-GBML ranking	Chi et al. ranking	PDFC ranking
IM	3	3	2
EC	9.5	13	1
SVMi	3	2	2
MC	3	5.5	3
CMC	1	1	4.5

to relate the influence of the imputation methods to the information contained in the dataset. In order to study the 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 (1972) observes the noise in the dataset. 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

$$\text{Wilson's noise} = \frac{\text{noise}}{\# \text{ instances in the dataset}}.$$

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 (Cover and Thomas 1991). Recently, the use of the MI measure in feature selection has become well known and seen to be successful (Kwak and Choi 2002a, b; Peng et al. 2005). The use of the MI measure for continuous attributes has been tackled by (Kwak and Choi 2002a), allowing us to compute the MI measure not only in nominal-valued datasets. 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 dataset with one imputation method in respect to the not imputed dataset. The average of these ratios will show us if the imputation of the dataset produces a gain in information:

$$\text{Avg. MI ratio} = \frac{\sum_{x_i \in X} \frac{MI_z(x_i)+1}{MI(x_i)+1}}{|X|}$$

where  $X$  is the set of input attributes,  $MI_z(i)$  represents the MI value of the  $i$ th attribute in the imputed dataset and  $MI(i)$  is the MI value of the  $i$ th input attribute in the not imputed dataset. 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

(Kwak and Choi 2002a) 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 dataset,  $N$  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  $\Sigma$  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 9, we have summarized the Wilson’s noise ratio values for the 21 datasets considered in our study. We must point out that the results of Wilson’s noise ratio are related to a given dataset. Hence, the characteristics of the proper data appear to determine the values of this measure.

In Table 10, we have summarized the average MI ratios for the 21 datasets. 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



**Table 9** Wilson's noise ratio values

Dataset	Imp. method	% Wilson's noise ratio	Dataset	Imp. method	% Wilson's noise ratio	Dataset	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	<b>16.0000</b>
	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
	SVM	50.0000		SVM	6.9307		SVM	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	<b>4.9505</b>		LLSI	18.6667
	EC	<b>33.3333</b>		EC	7.4257		EC	<b>16.0000</b>
WIS	MC	18.7500	WAT	MC	31.5068	MUS	MC	<b>0.0000</b>
	CMC	<b>12.5000</b>		CMC	<b>21.2329</b>		CMC	<b>0.0000</b>
	KNNI	<b>12.5000</b>		KNNI	27.3973		KNNI	<b>0.0000</b>
	WKNNI	<b>12.5000</b>		WKNNI	27.3973		WKNNI	<b>0.0000</b>
	KMI	<b>12.5000</b>		KMI	27.3973		KMI	<b>0.0000</b>
	FKMI	<b>12.5000</b>		FKMI	31.5068		FKMI	<b>0.0000</b>
	SVM	<b>12.5000</b>		SVM	23.9726		SVM	<b>0.0000</b>
	EM	<b>12.5000</b>		EM	46.5753		EM	<b>0.0000</b>
	SVDI	<b>12.5000</b>		SVDI	49.3151		SVDI	<b>0.0000</b>
	BPCA	<b>12.5000</b>		BPCA	26.0274		BPCA	<b>0.0000</b>
	LLSI	<b>12.5000</b>		LLSI	25.3425		LLSI	<b>0.0000</b>
	EC	<b>12.5000</b>		EC	22.6027		EC	<b>0.0000</b>
CRX	MC	18.9189	SPO	MC	27.2727	POS	MC	<b>33.3333</b>
	CMC	18.9189		CMC	<b>22.7273</b>		CMC	<b>33.3333</b>
	KNNI	21.6216		KNNI	27.2727		KNNI	<b>33.3333</b>
	WKNNI	21.6216		WKNNI	27.2727		WKNNI	<b>33.3333</b>
	KMI	21.6216		KMI	27.2727		KMI	<b>33.3333</b>
	FKMI	18.9189		FKMI	27.2727		FKMI	<b>33.3333</b>
	SVM	13.5135		SVM	27.2727		SVM	<b>33.3333</b>
	EM	32.4324		EM	36.3636		EM	<b>33.3333</b>
	SVDI	27.0270		SVDI	31.8182		SVDI	<b>33.3333</b>
	BPCA	21.6216		BPCA	27.2727		BPCA	<b>33.3333</b>
	LLSI	18.9189		LLSI	27.2727		LLSI	<b>33.3333</b>
	EC	<b>13.5135</b>		EC	27.2727		EC	<b>33.3333</b>
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
	SVM	55.5556		SVM	<b>21.2928</b>		SVM	44.4444
	EM	<b>44.4444</b>		EM	26.2357		EM	51.1111
	SVDI	<b>44.4444</b>		SVDI	22.4335		SVDI	48.8889
	BPCA	66.6667		BPCA	23.9544		BPCA	44.4444
	LLSI	66.6667		LLSI	24.7148		LLSI	<b>37.7778</b>
	EC	66.6667		EC	23.5741		EC	48.8889

**Table 9** continued

Dataset	Imp. method	% Wilson's noise ratio	Dataset	Imp. method	% Wilson's noise ratio	Dataset	Imp. method	% Wilson's noise ratio
AUT	MC	45.6522	HOC	MC	19.3906	SOY	MC	<b>2.4390</b>
	CMC	41.3043		CMC	<b>10.2493</b>		CMC	<b>2.4390</b>
	KNNI	41.3043		KNNI	20.2216		KNNI	<b>2.4390</b>
	WKNNI	41.3043		WKNNI	19.1136		WKNNI	<b>2.4390</b>
	KMI	41.3043		KMI	21.8837		KMI	<b>2.4390</b>
	FKMI	45.6522		FKMI	20.4986		FKMI	<b>2.4390</b>
	SVMi	43.4783		SVMi	20.2216		SVMi	<b>2.4390</b>
	EM	58.6957		EM	21.0526		EM	<b>2.4390</b>
	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	<b>2.4390</b>
	EC	<b>30.4348</b>		EC	20.7756		EC	<b>2.4390</b>
PRT	MC	71.0145	AUD	MC	38.7387	MAM	MC	21.3740
	CMC	<b>60.8696</b>		CMC	<b>32.8829</b>		CMC	<b>13.7405</b>
	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	<b>0.0000</b>	LUN	MC	80.0000	OZO	MC	4.8035
	CMC	<b>0.0000</b>		CMC	80.0000		CMC	<b>3.6390</b>
	KNNI	<b>0.0000</b>		KNNI	80.0000		KNNI	4.3668
	WKNNI	<b>0.0000</b>		WKNNI	80.0000		WKNNI	4.5124
	KMI	<b>0.0000</b>		KMI	80.0000		KMI	4.9491
	FKMI	<b>0.0000</b>		FKMI	80.0000		FKMI	4.0757
	SVMi	<b>0.0000</b>		SVMi	80.0000		SVMi	3.7846
	EM	<b>0.0000</b>		EM	<b>20.0000</b>		EM	4.8035
	SVDI	<b>0.0000</b>		SVDI	40.0000		SVDI	4.8035
	BPCA	<b>0.0000</b>		BPCA	80.0000		BPCA	4.3668
	LLSI	<b>0.0000</b>		LLSI	80.0000		LLSI	4.2213
	EC	<b>0.0000</b>		EC	80.0000		EC	4.8035

the MVs using such relationships and can afford improvements in the performance of the FRBCSs. 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 datasets are more or less close to each other.

If we count the methods with the lowest Wilson's noise ratios in each dataset in Table 9, 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 dataset in Table 10, the EC method has the highest ratio for 7 datasets and is therefore the first one. The CMC method has the highest ratio for 5 datasets and is the second one in this case. Considering the analysis of the previous Sect. 5.2 with these two methods:

**Table 10** Average mutual information ratio

Dataset	Imp. method	Avg. MI ratio	Dataset	Imp. Method	Avg. MI ratio	Dataset	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
	SVM	0.998365		SVM	0.908067		SVM	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	<b>1.122904</b>		LLSI	0.995464
EC	<b>1.000148</b>	EC	1.007843	EC	<b>1.024019</b>			
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
	SVM	0.999412		SVM	0.967356		SVM	0.981261
	EM	0.990030		EM	0.933846		EM	<b>1.142177</b>
	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	<b>1.000030</b>	EC	<b>1.027369</b>	EC	1.017366			
CRX	MC	1.000883	SPO	MC	0.997675	POS	MC	1.012293
	CMC	1.000966		CMC	<b>1.022247</b>		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
	SVM	0.981878		SVM	1.015835		SVM	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	<b>1.018007</b>
EC	<b>1.008718</b>	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	<b>0.998134</b>
	KMI	0.998709		KMI	1.016101		KMI	0.967169
	FKMI	0.998709		FKMI	1.020989		FKMI	0.983606
	SVM	0.998709		SVM	<b>1.542536</b>		SVM	0.987678
	EM	<b>1.013758</b>		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			

**Table 10** continued

Dataset	Imp. method	Avg. MI ratio	Dataset	Imp. Method	Avg. MI ratio	Dataset	Imp. method	Avg. MI ratio
AUT	MC	0.985610	HOC	MC	0.848649	SOY	MC	1.056652
	CMC	0.991113		CMC	<b>2.039992</b>		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
	SVM	0.991850		SVM	0.843456		SVM	<b>1.772589</b>
	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	<b>1.007652</b>		EC	0.911543		EC	1.222631
	PRT	MC		0.949896	AUD		MC	0.990711
CMC		<b>1.120006</b>	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	
SVM		1.038152	SVM	1.007958		SVM	<b>1.272993</b>	
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	<b>1.209608</b>		EC	1.269505	
DER		MC	1.000581	LUN		MC	0.996176	OZO
	CMC	<b>1.002406</b>	CMC		1.008333	CMC	<b>0.989156</b>	
	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	
	SVM	1.001566	SVM		1.006028	SVM	0.988297	
	EM	1.000016	EM		1.067844	EM	0.979977	
	SVDI	0.999691	SVDI		<b>1.076334</b>	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 EC method is the best method obtained for PDFC, and the third best for the rule Induction Learning methods while is one of the worst for Chi et al. and PDFC methods. Therefore, the TSK models seems to benefit more from those imputation methods which produce gain in the MI.
- The CMC method is the best method for the Mamdani models (Chi et al. and FH-GBML), and not very bad for PDFC. Mamdani FRBCSs benefit from the imputation method which induce less noise in the resultant imputed dataset.

Next, we rank all the imputation methods according to the values presented in Tables 9 and 10. In order to do so, we have calculated the average rankings of each imputation method for all the datasets, 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. 5.2. In Table 11 we have gathered together these average rankings, as well as their relative position in parentheses.

From the average rankings shown in Table 11, we can observe that the CMC method is the first for both rankings. The EC method is the second for the mutual information

**Table 11** 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	<b>3.79</b> (1)	<b>3.60</b> (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	<b>5.31</b> (3)	<b>3.86</b> (2)

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 the best results for the FRBCSs 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.

## 7 Concluding remarks

This study is a general comparison of FRBCSs not previously considered in MV studies. We have studied the use of imputation techniques for the analysis of three representative FRBCSs, presenting an analysis among imputation, do not impute and ignore cases with MVs. We have used a large bunch of datasets with real MVs to do so.

From the obtained results in Sect. 5.2, the particular analysis of the MVs treatment methods conditioned to the FRBCS nature is necessary. Thus, we can stress particular imputation algorithms based on the classification groups, as in the case of the CMC method for the Mamdami FRBCSs

and the EC method for the TSK models. Therefore, we can confirm the positive effect of the imputation methods and the FRBCS' behavior, and the presence of more suitable imputation methods for some particular FRBCS categories than others.

Moreover, we have analyzed the influence of the imputation methods in 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, which correspond to the best imputation methods observed for each FRBCS types.

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

## References

- Acuna E, Rodriguez C (2004) The treatment of missing values and its effect in the classifier accuracy. In: Banks D, House L, McMorris F, Arabie P, Gaul W (eds) Classification, clustering and data mining applications. Springer, Berlin, pp 639–648
- 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
- Barnard J, Meng X (1999) Applications of multiple imputation in medical studies: From AIDS to NHANES. *Stat Methods Med Res* 8(1):17–36
- Batista G, Monard M (2003) An analysis of four missing data treatment methods for supervised learning. *Appl Artif Intell* 17(5):519–533
- Berthold MR, Huber KP (1998) Missing values and learning of fuzzy rules. *Int J Uncertain, Fuzziness and Knowl-Based Syst* 6:171–178
- Chen Y, Wang JZ (2003) Support vector learning for fuzzy rule-based classification systems. *IEEE Trans on Fuzzy Systems* 11(6):716–728
- Chi Z, Yan H, Pham T (1996) Fuzzy algorithms with applications to image processing and pattern recognition. World Scientific
- Cover TM, Thomas JA (1991) Elements of Information Theory, 2nd edn. John Wiley
- Cristianini N, Shawe-Taylor J (2000) An introduction to support vector machines and other kernel-based learning methods. Cambridge University Press, New York
- Demšar J (2006) Statistical comparisons of classifiers over multiple data sets. *J Mach Learn Res* 7:1–30
- Dubois D, Prade H (1978) Operations on fuzzy numbers. *International Journal of Systems Sciences* 9:613–626
- 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



- 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
- 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) 9th international conference on knowledge-based & intelligent information & engineering systems (KES 2005), Springer, Lecture Notes in Computer Science, vol 3683, pp 581–587
- Gabriel TR, Berthold MR (2005) Missing values in fuzzy rule induction. In: Anderson G, Tunstel E (eds) 2005 IEEE conference on systems, man and cybernetics, IEEE Press
- 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
- García-Laencina P, Sancho-Gómez J, Figueiras-Vidal A (2009) Pattern classification with missing data: a review. *Neural Comput Appl* 9(1):1–12
- Gheys IA, Smith LS (2010) A neural network-based framework for the reconstruction of incomplete data sets. *Neurocomputing* 73(16–18):3039–3065
- Grzymala-Busse J, Goodwin L, Grzymala-Busse W, Zheng X (2005) Handling missing attribute values in preterm birth data sets. In: 10th international conference of rough sets and fuzzy sets and data mining and granular computing (RSFDGrC’5), pp 342–351
- 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*, Springer, lecture notes in computer science, vol 2005, pp 378–385
- Ishibuchi H, Nakashima T, Nii M (2004) Classification and modeling with linguistic information granules: advanced approaches to linguistic data mining. Springer-Verlag New York Inc.
- Ishibuchi H, Yamamoto T, Nakashima T (2005) Hybridization of fuzzy GBML approaches for pattern classification problems. *IEEE Trans Syst, Man Cybernet B* 35(2):359–365
- Hruschka Jr. ER, Hruschka ER, Ebecken NFF (2007) Bayesian networks for imputation in classification problems. *J Intell Inf Syst* 29(3):231–252
- 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
- Kuncheva L (2000) *Fuzzy classifier design*. Springer, Berlin
- Kwak N, Choi CH (2002a) Input feature selection by mutual information based on parzen window. *IEEE Trans Pattern Anal Mach Intell* 24(12):1667–1671
- Kwak N, Choi CH (2002b) Input feature selection for classification problems. *IEEE Trans Neural Netw* 13(1):143–159
- Li D, Deogun J, Spaulding W, Shuart B (2004) Towards missing data imputation: a study of fuzzy k-means clustering method. In: 4th international conference of rough sets and current trends in computing (RSCTC04), pp 573–579
- Little RJA, Rubin DB (1987) *Statistical Analysis with Missing Data*, 1st edn. Wiley series in probability and statistics. Wiley, New York
- 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 event covering method. *Neural Netw* 23:406–418
- 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) 19th Brazilian symposium on artificial intelligence (SBIA 2008), Springer, Lecture Notes in Computer Science, vol 5249, pp 217–226
- 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
- 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
- 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
- Pyle D (1999) *Data preparation for data mining*. Morgan Kaufmann
- Schafer JL (1997) *Analysis of incomplete multivariate data*. Chapman & Hall, London
- 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
- 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
- Vapnik VN (1998) *Statistical learning theory*. Wiley-Interscience
- Wang H, Wang S (2010) Mining incomplete survey data through classification. *Knowl Inf Syst* 24(2):221–233
- Wang LX, Mendel JM (1992) Generating fuzzy rules by learning from examples. *IEEE Trans Syst, Man, Cybernet* 25(2):353–361
- Wilson D (1972) Asymptotic properties of nearest neighbor rules using edited data. *IEEE Trans Syst, Man, Cybernet* 2(3):408–421
- Wong AKC, Chiu DKY (1987) Synthesizing statistical knowledge from incomplete mixed-mode data. *IEEE Trans Pattern Anal Mach Intell* 9(6):796–805