Pattern Recognition 47 (2014) 3941–3948

Contents lists available at ScienceDirect

Pattern Recognition

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

# Statistical computation of feature weighting schemes through data estimation for nearest neighbor classifiers

José A. Sáez [a,*], Joaquín Derrac [b], Julián Luengo [c], Francisco Herrera [a]

[a] Department of Computer Science and Artificial Intelligence, University of Granada, CITIC-UGR, Granada 18071, Spain
[b] School of Computer Science & Informatics, Cardiff University, Cardiff CF24 3AA, United Kingdom
[c] Department of Civil Engineering, LSI, University of Burgos, Burgos 09006, Spain

ARTICLE INFO

*Article history:*
Received 21 March 2013
Received in revised form
14 May 2014
Accepted 14 June 2014
Available online 24 June 2014

*Keywords:*
Feature weighting
Imputation methods
Nearest neighbor
Classification

ABSTRACT

The Nearest Neighbor rule is one of the most successful classifiers in machine learning. However, it is very sensitive to noisy, redundant and irrelevant features, which may cause its performance to deteriorate. Feature weighting methods try to overcome this problem by incorporating weights into the similarity function to increase or reduce the importance of each feature, according to how they behave in the classification task. This paper proposes a new feature weighting classifier, in which the computation of the weights is based on a novel idea combining imputation methods – used to estimate a new distribution of values for each feature based on the rest of the data – and the Kolmogorov–Smirnov nonparametric statistical test to measure the changes between the original and imputed distribution of values. This proposal is compared with classic and recent feature weighting methods. The experimental results show that our feature weighting scheme is very resilient to the choice of imputation method and is an effective way of improving the performance of the Nearest Neighbor classifier, outperforming the rest of the classifiers considered in the comparisons.



## 1. Introduction

The Nearest Neighbor (NN) classifier [1] is one of the most widely used methods in classification tasks due to its simplicity and good behavior in many real-world domains [2]. It is a nonparametric classifier which simply uses the full training data set to establish a classification rule, based on the most similar or nearest training instance to the query example.

The most frequently used similarity function for the NN classifier in the instance-based learning area is Euclidean distance [3]. However, redundant, irrelevant and highly correlated features may lead to erroneous similarities between the examples obtained and, therefore, to a deterioration in performance [4]. One way of overcoming this problem lies in modifying the similarity function, that is, the way in which the distances are computed. With this objective, weighting schemes can be applied in order to improve the similarity function, by introducing a weight for each of the features. High weights are assigned to those features that are helpful to classification and low weights are assigned to harmful or redundant features.

Feature Weighting methods [5] are able to enhance the NN classifier following the above procedure. By contrast to Feature Selection [6–9], the usage of weighting schemes provides the classifiers with a way of considering features *partially*, giving them some degree of importance in the classification task. This is usually preferred since weak, yet useful features may still be considered, instead of forcing the methods to either accept or completely ignore them. Many approaches using Feature Weighting have been proposed in the literature, some of which have focused on the NN classifier [10–12].

This paper proposes a novel approach for weighting features, based on the usage of imputation methods [13,14]. These are commonly employed to estimate those feature values in a data set that are unknown, formally known as missing values (MV) [15], using the rest of the data available. Therefore, imputation methods enable us to estimate a new distribution of the original data set, in which the distribution of each feature is conditioned to the rest of the features or all the data. These conditioned distributions of each feature can be compared with the original ones in order to detect the relevance of each feature, depending on the accuracy of the estimation for that feature performed by the imputation method.

The Kolmogorov–Smirnov statistic [16] may then be used to evaluate the differences between the original distribution of the features and that of the imputed ones. It is thus possible to measure how well the values of each feature can be predicted

* Corresponding author. Tel.: +34 958 240598; fax: +34 958 243317.
*E-mail addresses:* smja@decsai.ugr.es (J.A. Sáez), jderrac@decsai.ugr.es (J. Derrac), jluengo@ubu.es (J. Luengo), herrera@decsai.ugr.es (F. Herrera).

http://dx.doi.org/10.1016/j.patcog.2014.06.012
0031-3203/

using the rest of the data. This enables us to give more importance to those features with high changes between their original and estimated value distributions – these features keep most of the structural information of the data and are not easily predictable using the rest of the data, which reduces the effect of those features that are easily predictable, and which are therefore likely to be redundant.

The study is completed with an experimentation in which our proposal is compared with several classic and recent proposals of feature weighting, considering 25 supervised classification problems taken from the Keel-Dataset repository [17]. A web page with material complementary to this paper is available at http://sci2s.ugr.es/fw-imputation including the data sets used and the performance results of each classifier.

The rest of this paper is organized as follows. Section 2 introduces imputation and feature weighting methods. In Section 3 we describe our proposal. In Section 4 we present the experimental framework, and in Section 5 we analyze the results obtained. Finally, in Section 6 we enumerate some concluding remarks.

## 2. Preliminaries

This section introduces our proposal's main topics: imputation in Section 2.1 and feature weighting in Section 2.2.

### 2.1. Imputation methods for the estimation of values

Many real-world problems contain missing values as a result of, for example, manual data entry procedures or equipment errors. This poses a severe problem for machine learning applications, since most classifiers cannot work directly with incomplete data sets. Furthermore, MVs may cause different problems in a classification task [13]: (i) loss of efficiency, (ii) complications in handling and analyzing the data and (iii) bias resulting from differences between missing and complete data. Therefore, a preprocessing stage in which the data are prepared and cleaned is usually required [18].

Imputation methods [14,19] aim to predict a value for each MV. In most cases, the features of a data set are not independent of each other. Thus, through the identification of relationships among features, MVs can be determined. An advantage of this approach is that the MV treatment is independent of the learning algorithm used. Hence, the user is able to select the most appropriate imputation depending on the learning approach considered [13].

One of the simplest imputation methods is based on the NN rule: *k-NN Imputation* (KNNI). C4.5 or CN2 usually benefit from its usage [19]. Other approaches try to improve or complement its performance over various domains, for example, in [20] a *Support Vector Machine* (SVM) was used to fill in MVs (SVMI).

Other works are mostly focused on studying the behavior of several imputation methods in a specific scenario. For example, in [21], the authors induced MVs in several data sets. The prediction value – that is, the similarity of the imputed value to the originally removed one – of several imputation methods, such as *Regularized Expectation-Maximization* [22] or *Concept Most Common* (CMC) [23], and the accuracy obtained by several classifiers were studied. From the results, the authors stated that better prediction results do not imply better classification results. A similar approach was adopted in [14], in which the behavior of classifiers belonging to different paradigms, such as decision trees or instance-based learning methods, was studied over data sets with different levels of MVs.

All the aforementioned works have shown that imputation methods work properly when estimating missing values from the rest of the available data. They are therefore also suitable for use in our proposal.

### 2.2. Feature weighting in nearest neighbor classification

Data preparation [18,24] provides a number of ways to improve the performance of the NN classifier, such as Prototype Selection [25] or Feature Selection [6–9]. A different, yet powerful approach is Feature Weighting [5].

Feature Weighting methods can be included as a part of another type of more general methods: those based on adaptive distance measures [26–29]. These techniques try to learn distance metrics from the labeled examples of a problem in order to improve the classification performance. A reference work within this topic is, for example, that of Weinberger and Saul [26], in which the Mahanalobis distance metric is learned for *k*-nearest neighbor classification by semidefinite programming. The metric is trained in order that the *k*-nearest neighbors always belong to the same class while examples from different classes are separated by a large margin. On the other hand, the approach of [29] proposes a framework in which the metrics are parameterized by pairs of identical convolutional neural nets. Other works [27,28] consider schemes for locally adaptive distance metrics that change across the input space to overcome the bias problem of NN when working in high dimensions. In [27] a local linear discriminant analysis is used to compute neighborhoods, whereas in [28] a technique that computes a locally flexible metric by means of support vector machines is proposed.

The main objective of Feature Weighting methods is to reduce the sensitivity of the NN rule to redundant, irrelevant or noisy features. This is achieved by modifying its similarity function [4] with the inclusion of weights. These weights can be regarded as a measure of how useful a feature is with respect to the final classification task. The higher a weight is, the more influence the associated feature will have in the decision rule used to compute the classification of a given example. Therefore, an adequate scheme of weights could be used to highlight the best features of the domain of the problem, diminishing the impact of redundant, irrelevant and noisy ones. Thus, the accuracy of the classifier could be greatly improved if a proper selection of weights is made.

In the case of the NN classifier, most of the techniques developed to include Feature Weighting schemes have been focused on incorporating the weights in the distance measure, mainly to Euclidean distance (see Eq. (1), where $X$ and $Y$ are two instances and $M$ is the number of features that describes them). In spite of its simplicity, the usage of Euclidean distance has been preferred in many research approaches, since it is easy to optimize and shows a good discriminative power in most classification tasks. In fact, it is the most commonly used similarity measure in the instance based learning field [3].

$$d(X,Y) = \sqrt{\sum_{i=0}^{M} (x_i - y_i)^2} \tag{1}$$

Feature Weighting methods often extend this definition through the inclusion of weights associated with each feature ($W_i$, usually $W_i \in [0,1]$). These modify the way in which the distance measure is computed (Eq. (2)), increasing the relevance (the squared difference between feature's values) of those features with greater weights associated with them (near to 1.0).

$$d_w(X,Y) = \sqrt{\sum_{i=0}^{M} W_i \cdot (x_i - y_i)^2} \tag{2}$$

The application of this technique to the NN classifier has been widely addressed. To the best of our knowledge, the most complete study undertaken to this end can be found in [5], in which a review of several Feature Weighting methods for Lazy Learning algorithms [30] is presented (with most of them applied to improve the performance of the NN rule). In this review, Feature Weighting techniques are categorized by several dimensions, regarding the weight learning bias, the weight space (binary or continuous), the representation of features employed, their generality and their degree of employment of domain specific knowledge.

A wide range of classical Feature Weighting techniques are available in the literature, both classical (see [5] for a complete review) and recent [10,12]. The most well known compose the family of *Relief-based* algorithms.

The Relief algorithm [31] (which was originally a Feature Selection method [6]) has been widely studied and modified, producing several interesting variations of the original approach. Some of them [32,11] are based on ReliefF [33], which is the first adaptation of Relief as a Feature Weighting approach.

In addition to these approaches, Feature Weighting methods are also very useful when considered as a part of larger supervised learning schemes. In these approaches, Feature Weighting can be regarded as an improved version of Feature Selection (in fact, Feature Selection is a binary version of Feature Weighting, defining a weight of 1 if a feature is selected, or 0 if it is discarded). Again, if the weights scheme is properly chosen, Feature Weighting can play a decisive role in enhancing the performance of the NN classifier in these techniques [34].

## 3. A weighting algorithm based on feature differences after values imputation

This section describes the weighting method proposed, which is based on three main steps (see Fig. 1):

1. *Imputation of the data set* (Section 3.1): In this phase, an imputation method is used to build a new estimated data set $DS'$ from the original one $DS$.
2. *Computation of weights* (Section 3.2): The distribution of the values of each feature $f_i$ of $DS$ and the corresponding estimated feature $f_i'$ of $DS'$ are compared using the Kolmogorov–Smirnov statistical test. This enables the extraction of the $D_n^i$ statistic for each feature $f_i$.
3. *Construction of the classifier* (Section 3.3): Once the $D_n^i$ statistic is computed for each feature $i$, the NN classifier is used, incorporating a modified version of Euclidean distance. This version is based on a weighting scheme derived from the $D_n^i$ statistics.

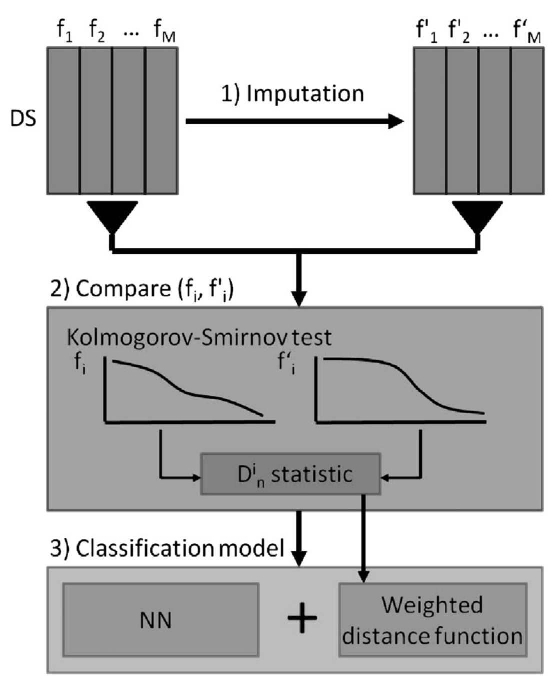


**Fig. 1.** Feature weighting method proposed.

The following sections describe each of these steps in depth. Section 3.1 is devoted to the imputation phase, whereas Section 3.2 describes the computation of the weights. Finally, Section 3.3 characterizes the classification model.

### 3.1. Imputation of the data set

The first step consists of creating a whole new estimated data set $DS'$ from the original one $DS$. In order to do this, an imputation method is used (in this paper we will consider KNNI [19], CMC [23] and SVMI [20], although other imputation methods may be chosen). If the original data set $DS$ is composed of the features $f_1, f_2, \ldots, f_M$, the imputed data set $DS'$ will be formed by the features $f'_1, f'_2, \ldots, f'_M$ whose values are obtained by the imputation method.

The procedure to obtain $DS'$ from $DS$ is represented in Algorithm 1. This is based on assuming iteratively that each feature value of each example of the data set $DS$, that is, $e(f_i)$, is missing (lines 2–5). Then, the imputation method $IM$ is used to predict a new value for that feature value (line 6). The new data set $DS'$ is obtained by repeating this process for each feature value, until the whole data set has been processed. Carrying out this process, it is possible to estimate a distribution of values for each feature, which is conditioned to the rest of the features or the totality of the data. The new data set $DS'$ will contain these conditioned distributions for each feature. This will allow us to check those features that are more difficult to predict with the rest of the features/data and contain the structural information of the data set, making them more important to the classification task.

**Algorithm 1.** Pseudocode of the first step of the method: imputation of the dataset.

**Input**: original dataset $DS$, imputation method $IM$.
**Output**: estimated dataset $DS'$.

```
1  Set DS' = ∅;
2  for each example e ∈ DS do
3  |  e' = null;
4  |  for each feature f_i do
5  |  |  Suppose e(f_i) as missing;
6  |  |  e'(f'_i) ⟵ Estimate the value for e(f_i) using IM over DS;
7  |  end
8  |  DS' ⟵ DS' ∪ {e'}
9  end
```

### 3.2. Computation of weights using the Kolmogorov–Smirnov test

The next step consists of measuring which features are most changed after the application of the imputation method. Given the nature of the imputation techniques, some features are expected to remain unchanged (or to present only small changes in their values' distribution) whereas other features may present a higher level of disruption when their imputed values are compared with the original ones. The Kolmogorov–Smirnov test [16] provides a way of measuring these changes. This test works by computing a statistic $D_n$, which can be regarded as a measure of how different two samples are.

The test is a nonparametric procedure for testing the equality of two continuous, one-dimensional probability distributions. It quantifies a distance between the empirical distribution functions of two samples. The null distribution of its statistic, $D_n$, is computed under the null hypothesis that the samples are drawn from the same distribution.

The main advantage of using the $D_n$ statistic (computed in the Kolmogorov–Smirnov test) instead of other simpler statistics such as the variance is that, for our purpose, which consists of measuring the similarity of two given distributions, shape measures used to compare the two distributions are more appropriate than other types of measures (such as dispersion measures in the case of the variance). Thus, when comparing two distributions, the changes in the variance do not provide enough information on how similar the two distributions are. Variances are only a measure of how the values of an attribute are concentrated around the mean, and is just one of the many factors that may be changed by distribution. However, the $D_n$ statistic contains the structural information that describes how the distribution has changed. This can be done by identifying where the higher or lower concentrations of values are (in the lowest values of the distribution, in the highest values, if there are several intervals with a higher concentration of values, etc.). Thus, the $D_n$ statistic is therefore much more representative than a simple comparison between the variances of the two distributions.

On the other hand, two samples of values with the same variance do not necessarily imply that both follow the same distribution (the same shape), or even that they have similar distributions. A simple example in which the variance does not work properly can be seen in regard to the property that makes the variance invariant to changes in the origin. Suppose two attributes: $A$ (real distribution of values of an attribute) and $A'$ (the distribution with the estimated values of that attribute). Assume that $A' = A + C$, where $C$ is a constant. Then, $variance(A) = variance(A')$. The two samples have the same variance, even though they obviously come from two different distributions and this fact is not detected using the variance. This problem is avoided if the $D_n$ statistic is employed.

Given two samples, $X$ and $Y$, and their empirical distribution functions $F_X$ and $F_Y$

$$F_X(x) = \frac{1}{n}\sum_{i=1}^{n} I_{X_i \le x}, \quad F_Y(x) = \frac{1}{n}\sum_{i=1}^{n} I_{Y_i \le x} \tag{3}$$

(where $I_{X_i \le x}$ is the indicator function, equal to 1 if $X_i \le x$ and equal to 0 otherwise) the Kolmogorov–Smirnov statistic is

$$D_n = \sup_x |F_X - F_Y| \tag{4}$$

Table 1 shows two toy samples (where two distributions of values $X = \{X_1, \ldots, X_n\}$ and $Y = \{Y_1, \ldots, Y_n\}$ with $n = 5$ are given), whereas Table 2 shows an example of the computation of the Kolmogorov–Smirnov statistic from them.

In the approach of this paper, the $D_n$ statistic provides a valuable way of estimating the degree of change undergone by a feature through the imputation process. By computing the $D_n$ statistic associated with the differences between both samples of the feature (original and imputed), it is possible to measure the greater degree of difference between the expected distribution of both samples. Hence, the greater $D_n$ value obtained, the more different the imputed version of the feature distribution will be (when compared with the original one).

**Table 1**
Two toy samples of size $n = 5$.

| |
|---|
| $X = \{X_1 = 0.01, X_2 = 0.11, X_3 = 0.12, X_4 = 0.22, X_5 = 0.85\}$ |
| $Y = \{Y_1 = 0.09, Y_2 = 0.41, Y_3 = 0.65, Y_4 = 0.73, Y_5 = 0.91\}$ |

**Table 2**
Example of the computation of the Kolmogorov–Smirnov statistic.

| x | $F_X$ | $F_Y$ | $\lvert F_X - F_Y \rvert$ | $\sup_x \lvert F_X - F_Y \rvert$ |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0.01 | 0.2 | 0 | 0.2 | 0.2 |
| 0.09 | 0.2 | 0.2 | 0 | 0.2 |
| 0.11 | 0.2 | 0.2 | 0.2 | 0.2 |
| 0.12 | 0.6 | 0.2 | 0.4 | 0.4 |
| 0.22 | 0.8 | 0.2 | 0.6 | 0.6 |
| 0.41 | 0.8 | 0.4 | 0.4 | 0.6 |
| 0.65 | 0.8 | 0.6 | 0.2 | 0.6 |
| 0.73 | 0.8 | 0.8 | 0 | 0.6 |
| 0.85 | 1.0 | 0.8 | 0.2 | 0.6 |
| 0.91 | 1.0 | 1.0 | 0 | 0.6 |
| $D_n$ | 0.6 | | | |

The $D_n$ statistic can be easily transformed into a weight. Since $D_n \in [0, 1]$, features with a lower value of $D_n$ (near to 0.0) it will have little influence on the computation of the similarity function of the NN rule, whereas features with a higher value of $D_n$ (near to 1.0) will be the most influential when computing the distance between two examples. Defining the statistical $D_n^i$ for the feature $i$ as

$$D_n^i = \text{Kolmogorov–Smirnov}(f_i, f_i') \quad \forall i, f_i \in \mathcal{A}, f_i' \in \mathcal{A}' \tag{5}$$

(where $\mathcal{A}$ denotes the set of features of the original data set $DS$ and $\mathcal{A}'$ denotes the set of features imputed in $DS'$), then the weights $W_i \in [0, 1]$ computed for a feature $f_i \in \mathcal{A}$ are

$$W_i = D_n^i \Big/ \sum_{j=1}^{M} D_n^j \tag{6}$$

### 3.3. Final classification model

The final classifier considers NN with the weighted Euclidean distance (Eq. (2)) and the weights computed throughout the Kolmogorov–Smirnov statistic (Eq. (6)).

Considering weights computed from the $D_n$ statistic, we aim to highlight the effect that changing features have on the computation of the distance. These features, with a larger associated $D_n$ value, will be those poorly estimated by the imputation method (whose sample distribution differs greatly if the original and imputed versions are compared). They are preferred since they keep most of the structural information of the data, and are the key features describing the data set (they cannot be properly estimated using the rest of the data).

By contrast, features with a small $D_n$ value will be those whose sample distribution has not been changed after the application of the imputation method. Since these features are easily estimated when the rest of the data is available (the imputation method can recover their values properly), they are not preferred in the final computation of the distance, and thus a lower weight is assigned to them.

## 4. Experimental framework

This section presents the framework of the experimental study conducted. The imputation methods considered in the previous section are presented in Section 4.1, whereas Section 4.2 is devoted to the feature weighting methods used. Section 4.3

describes the data sets employed. Finally, Section 4.4 describes the methodology followed to analyze the results.

### 4.1. Imputation methods

The proposal described in this paper allows us to include any standard imputation method. For the sake of generality, we have chosen to test the behavior using three different imputation techniques, well-known representatives of the field [13,19]:

1. *KNNI* [19]: Based on the $k$-NN algorithm, every time an MV is found in a current example, KNNI computes the $k$ nearest neighbors and their average value is imputed. KNNI also uses the Euclidean distance as a similarity function.
2. *CMC* [23]: This method replaces the MVs by the average of all the values of the corresponding feature considering only the examples with the same class as the example to be imputed.
3. *SVMI* [20]: This is an SVM regression-based algorithm developed to fill in MVs. It works by firstly selecting the examples in which there are no missing feature values. In the next step, the method sets one of the input features, some of the values of which are missing, as the decision feature, and the decision feature as the input feature. Finally, an SVM for regression is used to predict the new decision feature.

The parameter setup used for their execution is presented in Table 3. Each imputation method considered will lead to a different feature weighting classifier. Throughout the study, we will denote them as FW-KNNI, FW-CMC and FW-SVMI.

### 4.2. Feature weighting methods for NN

In order to check the performance of the approach proposed, the following feature weighting algorithms for nearest neighbor classification as comparison methods have been chosen:

1. *NN* [1]: The NN rule is used as a baseline limit of performance which most of the methods should supersede.
2. *CW* [10]: A gradient descent based algorithm developed with the aim of minimizing a performance index that is an approximation of the leave one out error over the training set. In this approach, weights are obtained for each combination of feature and class, that is, the set of weights is different depending on the class of each training example.
3. *MI* [5]: Mutual Information (MI) between features can be used successfully as a weighting factor for NN based algorithms. This method was marked as the best preset FW method in [5].
4. *ReliefF* [33]: The first Relief-based method adapted to perform the FW process. By contrast to the original Relief method, weights computed in ReliefF are not binarized to 0,1. Instead, they are used as final weights for the NN classifier. This method was noted as the best performance-based FW method in [5].
5. *IRelief* [11]: A multiclass, iterative extension of Relief. The objective function of the iterative process aims at reducing the distances between each example and its nearest hit (nearest training example of the same class) and increasing the distances between each example and its nearest enemy (nearest training example of another class).

**Table 3**
Parameter specification for the imputation methods.

| Algorithm | Ref. | Parameters |
|---|---|---|
| KNNI | [19] | $k$ value: 10 |
| CMC | [23] | It has no parameters to be fixed |
| SVMI | [20] | Kernel type: RBF, C: 1.0, RBF-$\gamma$: 1.0 |

**Table 4**
Parameter specification for the classifiers of the study.

| Algorithm | Ref. | Parameters |
|---|---|---|
| NN | [1] | It has no parameters to be fixed |
| CW | [10] | $\beta$: Best in $[0.125, 128]$, $\mu$: Best in $[0.001, 0.1]$, $\epsilon$: 0.001, Iterations: 1000 |
| MI | [5] | It has no parameters to be fixed |
| ReliefF | [33] | $K$ value: Best in $[1, 20]$ |
| IRelief | [11] | Maximum iterations: 100, $\epsilon$: 0.00001, $\sigma$: Best in $[0.001, 1000]$ |

**Table 5**
Data sets employed in the experimentation.

| Data set | #EXA | #FEA | #CLA | Data set | #EXA | #FEA | #CLA |
|---|---|---|---|---|---|---|---|
| banana | 5300 | 2 | 2 | pima | 768 | 8 | 2 |
| bands | 365 | 19 | 2 | satimage | 6435 | 36 | 7 |
| bupa | 345 | 6 | 2 | sonar | 208 | 60 | 2 |
| dermatology | 358 | 34 | 6 | tae | 151 | 5 | 3 |
| ecoli | 336 | 7 | 8 | texture | 5500 | 40 | 11 |
| heart | 270 | 13 | 2 | vowel | 990 | 13 | 11 |
| hepatitis | 80 | 19 | 2 | wdbc | 569 | 30 | 2 |
| ionosphere | 351 | 33 | 2 | wine | 178 | 13 | 3 |
| iris | 150 | 4 | 3 | wq-red | 1599 | 11 | 11 |
| led7digit | 500 | 7 | 10 | wq-white | 4898 | 11 | 11 |
| mov-libras | 360 | 90 | 15 | wisconsin | 683 | 9 | 2 |
| newthyroid | 215 | 5 | 3 | yeast | 1484 | 8 | 10 |
| phoneme | 5404 | 5 | 2 | | | | |

Table 4 summarizes the parameter setup used for the feature weighting methods in the experimental study, which was used in the reference in which the methods were originally described.

### 4.3. Data sets

The experimentation considers 25 data sets from the KEEL-Dataset repository [17]. They are described in Table 5, where #EXA refers to the number of examples, #FEA to the number of numeric features and #CLA to the number of classes.

For data sets containing missing values (such as *bands* or *dermatology*), the examples with missing values were removed from the data sets before their usage and thus all the attribute values of the data sets considered are known. In this way, the percentage of missing values of each data set does not influence the results or conclusions obtained and it does not harm the methods that are not specially designed to deal with them. Therefore, the only missing values considered in this paper are those assumed during the execution of Algorithm 1 in order to build the new estimated distribution of values.

### 4.4. Methodology of analysis

The performance estimation of each classifier on each data set is obtained by means of 3 runs of a 10-fold *distribution optimally balanced stratified cross-validation* (DOB-SCV) [35], averaging its test accuracy results. The usage of this partitioning reduces the negative effects of both prior probability and covariate shifts [36] when classifier performance is estimated with cross-validation schemes. The results with the standard cross-validation can be found on the web page of this paper.

Statistical comparisons of the data sets considered will be also performed. Wilcoxon's test [37] will be applied to study the differences among the proposals of this paper and also between

**Table 6**
Test accuracy results.

| Data set | CW | MI | ReliefF | IRelief | NN | FW-KNNI | FW-CMC | FW-SVMI |
|---|---|---|---|---|---|---|---|---|
| banana | 61.59 | 59.51 | 87.7 | **88.00** | 87.87 | 87.91 | 87.79 | 87.60 |
| bands | **72.31** | 45.27 | 65.99 | 36.97 | 72.04 | 72.04 | 69.02 | 69.85 |
| bupa | 62.36 | 42.02 | 59.09 | 55.35 | 62.36 | **64.11** | 64.09 | 63.50 |
| dermatology | 95.18 | **97.19** | 96.9 | 93.22 | 94.9 | 94.62 | 95.75 | 95.21 |
| ecoli | 80.09 | 78.88 | 70.69 | 76.88 | 80.09 | 79.77 | 80.09 | **80.67** |
| heart | 76.3 | **82.96** | 78.89 | 78.89 | 74.81 | 73.7 | 75.19 | 75.19 |
| hepatitis | 82.94 | 81.27 | 80.26 | **85.65** | 82.94 | 85.62 | 81.83 | 82.94 |
| ionosphere | 85.96 | 87.4 | 90.26 | **91.11** | 87.11 | 88.55 | 87.09 | 87.67 |
| iris | **96.00** | 83.33 | 95.33 | 94.67 | 95.33 | 95.33 | **96.00** | 94.00 |
| led7digit | 44.88 | 51.55 | 51.6 | 51.62 | 44.88 | **52.45** | 51.55 | 52.03 |
| mov-libras | 82.81 | 69.85 | 25.6 | 84.1 | 82.81 | 85.73 | **85.95** | 85.51 |
| newthyroid | 97.19 | 94.35 | **98.59** | 95.32 | 97.19 | 96.26 | 97.19 | 96.71 |
| phoneme | 90.43 | 76.85 | 68.24 | 72.63 | 90.41 | 91.04 | 91.06 | **91.08** |
| pima | 70.45 | 69.13 | 63.02 | 66.28 | 70.45 | **70.97** | 70.71 | 70.19 |
| satimage | **90.94** | 90.46 | 90.89 | 90.64 | 90.88 | 90.71 | 90.71 | **90.94** |
| sonar | 86.52 | 73.13 | 84.99 | **87.52** | 86.52 | 86.97 | 86.55 | 86.02 |
| tae | 42.07 | 31.81 | 28.63 | **69.42** | 42.07 | 65.55 | 65.55 | 65.55 |
| texture | **99.15** | 98 | 99.09 | 98.8 | **99.15** | **99.15** | 99.07 | **99.15** |
| vowel | 99.39 | 80.51 | 98.89 | 99.19 | 99.39 | **99.49** | 99.39 | 99.39 |
| wdbc | 95.97 | **96.14** | 93.46 | 95.26 | 95.96 | 94.91 | 95.8 | 95.97 |
| wine | 95.58 | 97.84 | **98.36** | 97.25 | 95.58 | 95.58 | 96.14 | 96.69 |
| wq-red | 53.72 | 48.96 | 66 | 63.76 | 53.66 | **66.19** | 65.74 | 65.55 |
| wq-white | 50.19 | 54.06 | 51.1 | 20.89 | 50.17 | 66.67 | **67.36** | 67.04 |
| wisconsin | 95.61 | 96.49 | **96.78** | 96.49 | 96.04 | 96.04 | 95.75 | 96.05 |
| yeast | 51.68 | 37.95 | 43.03 | 49.3 | 51.55 | 53.71 | **54.05** | 53.57 |
| Average | 78.37 | 73 | 75.34 | 77.57 | 79.37 | **82.12** | 81.98 | 81.92 |
| Best result (out of 25) | 4 | 3 | 3 | 5 | 1 | 6 | 4 | 4 |

each of these proposals and NN using the Euclidean distance. Regarding the comparison among feature weighting methods, the results of the Friedman Aligned test [38] and the Finner procedure [39] will be computed. Comparisons with other tests, such as the Holm test [40], may be found on the web page of this paper. More information about these statistical procedures can be found at http://sci2s.ugr.es/sicidm/.

## 5. Analysis of results

This section presents the analysis of the results obtained. Table 6 shows the test accuracy obtained by each classifier on each data set. The best results for each data set are highlighted in bold. From this table, several remarks can be made:

- The method obtaining the best results in most single data sets is FW-KNNI (in 6 of the 25 data sets). It is followed by IRelief (5 data sets), FW-CMC, FW-SVMI and CW (4 data sets), MI and ReliefF (3 data sets) and NN (1 data set).
- Even though IReliefF or CW obtain the best results in a certain number of data sets – 5 and 4 respectively –, they show a variable performance for different problems. For instance, in data sets such as *banana* and *tae*, CW's results are very far from the results obtained by the best performing methods in these data sets. The same occurs for IRelief – in *bands*, *phoneme* and *wq-white* – whereas this issue is not very remarkable with regard to any of the other proposals of this paper. This fact shows that our methods are generally more robust than those of the rest of the algorithms included in the comparison.
- Regardless of the imputation method selected, our approaches usually obtain results close to those of the best performing method in each data set. Moreover, all of them obtain better accuracy on average than the comparison methods over the 25 problems.

**Table 7**
Wilcoxon's comparison of the proposed methods.

| | FW-KNNI | | | FW-CMC | | | FW-SVMI | | |
|---|---|---|---|---|---|---|---|---|---|
| Method | R+ | R− | $p$-value | R+ | R− | $p$-value | R+ | R− | $p$-value |
| FW-KNNI | – | – | – | 159 | 166 | 1.0000 | 191.5 | 133.5 | 0.4273 |
| FW-CMC | 166 | 159 | 0.9140 | – | – | – | 154 | 146 | 0.8970 |
| FW-SVMI | 133.5 | 191.5 | 1.0000 | 146 | 154 | 1.0000 | – | – | – |

To add depth to the analysis of the results, several statistical comparisons are performed below, studying the differences among the proposals of this paper, their comparison with NN and also with the rest of the feature weighting methods.

*Comparison among the feature weighting methods based on imputation*: The results of the three proposals of this paper (FW-KNNI, FW-CMC and FW-SVMI) shown in Table 6 are quite similar. In order to study whether there are statistical differences among them, Wilcoxon's test has been performed – see Table 7. In this table, the classifier of each row is established as the control method for the statistical test and its ranks (R+), the ranks in favor of the method of the column (R−) and the $p$-value associated are shown. From the high $p$-values obtained in these comparisons, one can conclude that statistical differences among the three proposals do not exist. This fact shows the robustness of the proposal independent of the imputation method chosen. Therefore, the good behavior of the approach is due to the strategy for obtaining the weights, which combines imputation methods and the Kolmogorov–Smirnov test; the concrete imputation method employed does not influence the results so much.

*Comparison with NN*: Table 8 shows the results of applying Wilcoxon's test to each of the proposals performed and NN. As the table shows, every proposal is statistically better than NN due to the low $p$-values obtained – all are lower than 0.05. This shows

**Table 8**
Wilcoxon's comparison of the proposed methods with NN.

| Methods | R+ | R− | *p*-value |
|---|---|---|---|
| FW-KNNI vs NN | 229 | 71 | 0.0229 |
| FW-CMC vs NN | 224.5 | 75.5 | 0.0327 |
| FW-SVMI vs NN | 228.5 | 71.5 | 0.0239 |

**Table 9**
Statistical comparison among feature weighting methods.

| | FW-KNNI | | FW-CMC | | FW-SVMI | |
|---|---|---|---|---|---|---|
| Method | Rank | $p_{Finn}$ | Rank | $p_{Finn}$ | Rank | $p_{Finn}$ |
| Imputation | 42.90 | – | 43.78 | – | 43.74 | – |
| CW | 60.36 | 0.0884 | 60.08 | 0.1117 | 59.94 | 0.1139 |
| MI | 84.30 | 0.0002 | 83.64 | 0.0004 | 83.90 | 0.0004 |
| ReliefF | 65.24 | 0.0576 | 65.26 | 0.0708 | 64.82 | 0.0778 |
| IRelief | 62.20 | 0.0787 | 62.24 | 0.0943 | 62.60 | 0.0866 |
| $p_{FA}$ | 0.0003 | | 0.0003 | | 0.0003 | |

that the application of our approach to feature weighting improves the performance of the NN classifier significantly, regardless of the specific imputation method chosen.

*Comparison among feature weighting methods*: Table 9 presents the statistical comparison performed for each proposal (FW-KNNI, FW-CMC and FW-SVMI). Each proposal is independently compared with the rest of the feature weighting methods since we have already confirmed that there are no significant differences among our three approaches (see Table 7). The ranks obtained by the Friedman Aligned procedure (Rank column), which represent the effectiveness associated with each algorithm, and the *p*-value related to the significance of the differences found by this test ($p_{FA}$ row) are shown. The $p_{Finn}$ column shows the adjusted *p*-value computed by the Finner test.

Looking at Table 9, we can observe that:

- The average ranks obtained by our proposals are the best (the lowest) and they are notably differentiated from the ranks of the rest of the methods.
- These are followed by CW, IRelief and ReliefF with very close ranks among them. MI obtains the highest rank.
- The *p*-values of the Friedman Aligned test are very low in every case, meaning that the differences found among the methods are very significant.
- The *p*-values obtained with the Finner procedure when comparing FW-KNNI, FW-CMC and FW-SVMI with the comparison algorithms are very low. The differences found are always significant (lower than 0.1), except in the case of FW-CMC and FW-SVMI with CW, in which the *p*-value obtained is still very low.

From the results of Tables 6– 9, it is possible to conclude that the proposals presented in this paper perform better than the rest of the feature weighting methods considered. They are also able to improve the performance of the NN classifier. Even though they do not obtain the best results in a large number of single data sets, the statistical tests illustrate the improvement of performance achieved by our approaches, showing a great robustness and a good behavior in most of the data sets. The comparison among our three proposals does not show statistical differences, suggesting that the strategy for obtaining the weights performs accurately independent of the concrete imputation method employed.

## 6. Conclusions

In this paper we have proposed a new scheme for feature weighting developed to improve the performance of the NN classifier, in which the weights are computed by combining imputation methods and the Kolmogorov–Smirnov statistic. From the experimental results it is possible to conclude that our feature weighting scheme is not very sensitive to the selection of the imputation method, since the results obtained in every case are quite similar regardless of the specific imputation technique chosen, and statistical differences among them have not been found.

The results obtained show that all our approaches enhance the performance of NN to a greater degree than the rest of the feature weighting methods analyzed. They also show a robust behavior in several domains, in contrast to the rest of the classifiers, which demonstrate a variable performance when different data sets are considered. The statistical analysis performed confirms our conclusions. The results with standard cross-validation provide similar conclusions to those shown here (see the results at http://sci2s.ugr.es/fw-imputation).

## Conflict of interest

None declared.


## Acknowledgments

Supported by the Projects TIN2011-28488, P10-TIC-06858 and P11-TIC-9704. J.A. Sáez holds an FPU grant from the Spanish Ministry of Education and Science (reference AP2009-2930).


## References


[1] T. Cover, P. Hart, Nearest neighbor pattern classification, IEEE Trans. Inf. Theory 13 (1967) 21–27.
[2] X. Wu, V. Kumar, J. Ross Quinlan, J. Ghosh, Q. Yang, H. Motoda, G.J. McLachlan, A. Ng, B. Liu, P.S. Yu, Z.-H. Zhou, M. Steinbach, D.J. Hand, D. Steinberg, Top 10 algorithms in data mining, Knowl. Inf. Syst. 14 (1) (2007) 1–37.
[3] D.W. Aha, D. Kibler, M.K. Albert, Instance-based learning algorithms, Mach. Learn. 6 (1991) 37–66.
[4] Y. Chen, E.K. Garcia, M.R. Gupta, A. Rahimi, L. Cazzanti, Similarity-based classification: concepts and algorithms, J. Mach. Learn. Res. 10 (2009) 747–776.
[5] D. Wettschereck, D.W. Aha, T. Mohri, A review and empirical evaluation of feature weighting methods for a class of lazy learning algorithms, Artif. Intell. Rev. 11 (1997) 273–314.
[6] H. Liu, H. Motoda (Eds.), Computational methods of Feature Selection, Chapman & Hall/CRC Data Mining and Knowledge Discovery Series, Chapman & Hall/CRC, London, 2007.
[7] B. Xue, M. Zhang, W. Browne, Particle swarm optimisation for feature selection in classification: novel initialisation and updating mechanisms, Appl. Soft Comput. J. 18 (2014) 261–276.
[8] J. Kersten, Simultaneous feature selection and Gaussian mixture model estimation for supervised classification problems, Pattern Recognit. 47 (8) (2014) 2582–2595.
[9] D. Liu, H. Qian, G. Dai, Z. Zhang, An iterative SVM approach to feature selection and classification in high-dimensional datasets, Pattern Recognit. 46 (9) (2013) 2531–2537.
[10] R. Paredes, E. Vidal, Learning weighted metrics to minimize nearest-neighbor classification error, IEEE Trans. Pattern Anal. Mach. Intell. 28 (7) (2006) 1100–1110.
[11] Y. Sun, Iterative relief for feature weighting: algorithms, theories, and applications, IEEE Trans. Pattern Anal. Mach. Intell. 29 (6) (2007) 1035–1051.
[12] F. Fernández, P. Isasi, Local feature weighting in nearest prototype classification, IEEE Trans. Neural Netw. 19 (1) (2008) 40–53.
[13] J. Luengo, S. García, F. Herrera, On the choice of the best imputation methods for missing values considering three groups of classification methods, Knowl. Inf. Syst. 32 (1) (2012) 77–108.
[14] A. Farhangfar, L. Kurgan, J. Dy, Impact of imputation of missing values on classification error for discrete data, Pattern Recognit. 41 (12) (2008) 3692–3705.
[15] M. Juhola, J. Laurikkala, Missing values: how many can they be to preserve classification reliability? Artif. Intell. Rev. 40 (3) (2013) 231–245.

[16] N.V. Smirnov, Estimate of deviation between empirical distribution functions in two independent samples (in Russian), Bull. Mosc. Univ. 2 (1939) 3–16.
[17] J. Alcalá-Fdez, A. Fernández, J. Luengo, J. Derrac, S. García, L. Sánchez, F. Herrera, Keel data-mining software tool: data set repository, integration of algorithms and experimental analysis framework, J. Multiple-Valued Logic Soft Comput. 17 (2–3).
[18] D. Pyle, Data Preparation for Data Mining, Morgan Kaufmann, Los Altos, 1999.
[19] G.E.A.P.A. Batista, M.C. Monard, An analysis of four missing data treatment methods for supervised learning, Appl. Artif. Intell. 17 (5–6) (2003) 519–533.
[20] H. Feng, G. Chen, C. Yin, B. Yang, Y. Chen, A SVM regression based approach to filling in missing values, in: Lecture Notes in Computer Science, vol. 3683, Springer, 2005.
[21] E.R. Hruschka Jr., E.R. Hruschka, N.F.F. Ebecken, Bayesian networks for imputation in classification problems, J. Intell. Inf. Syst. 29 (3) (2007) 231–252.
[22] T. Schneider, Analysis of incomplete climate data: estimation of mean values and covariance matrices and imputation of missing values, J. Clim. 14 (5) (2001) 853–871.
[23] J.W. Grzymala-Busse, L.K. Goodwin, W.J. Grzymala-Busse, X. Zheng, Handling missing attribute values in preterm Birth data sets, in: Lecture Notes in Computer Science, vol. 3642, Springer, 2005.
[24] J. Lara, D. Lizcano, M. Martínez, J. Pazos, Data preparation for KDD through automatic reasoning based on description logic, Inf. Syst. 44 (2014) 54–72.
[25] S. García, J. Derrac, J.R. Cano, F. Herrera, Prototype selection for nearest neighbor classification: taxonomy and empirical study, IEEE Trans. Pattern Anal. Mach. Intell. 34 (3) (2012) 417–435.
[26] K. Weinberger, L. Saul, Distance metric learning for large margin nearest neighbor classification, J. Mach. Learn. Res. 10 (2009) 207–244.
[27] T. Hastie, R. Tibshirani, Discriminant adaptive nearest neighbor classification, IEEE Trans. Pattern Anal. Mach. Intell. 18 (6) (1996) 607–616.
[28] C. Domeniconi, D. Gunopulos, J. Peng, Large margin nearest neighbor classifiers, IEEE Trans. Neural Netw. 16 (4) (2005) 899–909.
[29] S. Chopra, R. Hadsell, Y. LeCun, Learning a similiarty metric discriminatively with application to face verification, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR-05), San Diego, CA, vol. 1, 2005, pp. 539–546.
[30] D.W. Aha (Ed.), Lazy Learning, Springer, New York, 1997.
[31] K. Kira, L.A. Rendell, A practical approach to feature selection, in: Proceedings of the Ninth International Conference on Machine Learning, Aberdeen, Scotland, Morgan Kaufmann, 1992, pp. 249–256.
[32] M.R. Sikonja, I. Kononenko, Theoretical and empirical analysis of Relief and ReliefF, Mach. Learn. 53 (1–2) (2003) 23–69.
[33] I. Kononenko, Estimating attributes: analysis and extensions of RELIEF, in: Proceedings of the 1994 European Conference on Machine Learning, Catania, Italy, Springer Verlag, 1994, pp. 171–182.
[34] J. Derrac, I. Triguero, S. García, F. Herrera, Integrating instance selection, instance weighting and feature weighting for nearest neighbor classifiers by co-evolutionary algorithms, IEEE Trans. Syst. Man, Cybern. Part B: Cybern. 42 (5) (2012) 1383–1397.
[35] J.G. Moreno-Torres, J.A. Sáez, F. Herrera, Study on the impact of partition-induced dataset shift on k-fold cross-validation, IEEE Trans. Neural Netw. Learn. Syst. 23 (8) (2012) 1304–1312.
[36] J.G. Moreno-Torres, T. Raeder, R. Alaiz-Rodríguez, N.V. Chawla, F. Herrera, A unifying view on dataset shift in classification, Pattern Recognit. 45 (1) (2012) 521–530.
[37] F. Wilcoxon, Individual comparisons by ranking methods, Biometr. Bull. 1 (6) (1945) 80–83.
[38] S. García, A. Fernández, J. Luengo, F. Herrera, Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: experimental analysis of power, Inf. Sci. 180 (2010) 2044–2064.
[39] H. Finner, On a monotonicity problem in step-down multiple test procedures, J. Am. Stat. Assoc. 88 (1993) 920–923.
[40] S. Holm, A simple sequentially rejective multiple test procedure, Scand. J. Stat. 6 (1979) 65–70.

**José A. Sáez** received his M.Sc. in Computer Science from the University of Granada (Granada, Spain) in 2009. He is currently a Ph.D. student in the Department of Computer Science and Artificial Intelligence in the University of Granada. His main research interests include noisy data in classification, discretization methods and imbalanced learning.

**Joaquín Derrac** received the M.Sc. and Ph.D. in computer science from the University of Granada, Granada, Spain, in 2008 and 2013, respectively. His research interests include data mining, data reduction, nearest neighbor classifiers, statistical inference and evolutionary algorithms.

**Julián Luengo** received the M.S. in computer science and the Ph.D. degree from the University of Granada, Granada, Spain, in 2006 and 2011, respectively. His research interests include machine learning and data mining, data preparation in knowledge discovery and data mining, missing values, data complexity and fuzzy systems.

**Francisco Herrera** received his M.Sc. in Mathematics in 1988 and Ph.D. 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 been the supervisor of 30 Ph.D. students. He has published more than 260 papers in international journals. He is coauthor of the book "Genetic Fuzzy Systems: Evolutionary Tuning and Learning of Fuzzy Knowledge Bases" (World Scientific, 2001).

He currently acts as Editor in Chief of the international journals "Information Fusion" (Elsevier) and "Progress in Artificial Intelligence (Springer). He acts as an area editor of the International Journal of Computational Intelligence Systems and associated editor of the journals: IEEE Transactions on Fuzzy Systems, Information Sciences, Knowledge and Information Systems, Advances in Fuzzy Systems, and International Journal of Applied Metaheuristics Computing; and he serves as a member of several journal editorial boards, among others: Fuzzy Sets and Systems, Applied Intelligence, Information Fusion, Knowledge-Based Systems, Evolutionary Intelligence, International Journal of Hybrid Intelligent Systems, Memetic Computation, and Swarm and Evolutionary Computation.

He received the following honors and awards: ECCAI Fellow 2009, IFSA Fellow 2013, 2010 Spanish National Award on Computer Science ARITMEL to the "Spanish Engineer on Computer Science", International Cajastur "Mamdani" Prize for Soft Computing (Fourth Edition, 2010), IEEE Transactions on Fuzzy System Outstanding 2008 Paper Award (bestowed in 2011), 2011 Lotfi A. Zadeh Prize Best paper Award of the International Fuzzy Systems Association, and 2013 AEPIA Award to a scientific career in Artificial Intelligence (September 2013).

His current research interests include computing with words and decision making, bibliometrics, data mining, data preparation, instance selection and generation, imperfect data, fuzzy rule based systems, genetic fuzzy systems, imbalanced classification, knowledge extraction based on evolutionary algorithms, memetic algorithms and genetic algorithms, biometrics, cloud computing and big data.