Information Sciences 186 (2012) 73–92

Contents lists available at SciVerse ScienceDirect

Information Sciences

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

Enhancing evolutionary instance selection algorithms by means of fuzzy rough set based feature selection

Joaquín Derrac [a,*], Chris Cornelis [b], Salvador García [c], Francisco Herrera [a]

[a] Dept. of Computer Science and Artificial Intelligence, CITIC-UGR, Research Center on Information and Communications Technology, University of Granada, 18071 Granada, Spain
[b] Dept. of Applied Mathematics and Computer Science, Ghent University, Gent, Belgium
[c] Dept. of Computer Science, University of Jaén, 23071 Jaén, Spain

ARTICLE INFO

Article history:
Received 9 October 2010
Received in revised form 10 June 2011
Accepted 18 September 2011
Available online 29 September 2011

Keywords:
Instance selection
Feature selection
Rough sets
Evolutionary algorithms
Nearest neighbor

ABSTRACT

In recent years, fuzzy rough set theory has emerged as a suitable tool for performing feature selection. Fuzzy rough feature selection enables us to analyze the discernibility of the attributes, highlighting the most attractive features in the construction of classifiers. However, its results can be enhanced even more if other data reduction techniques, such as instance selection, are considered.

In this work, a hybrid evolutionary algorithm for data reduction, using both instance and feature selection, is presented. A global process of instance selection, carried out by a steady-state genetic algorithm, is combined with a fuzzy rough set based feature selection process, which searches for the most interesting features to enhance both the evolutionary search process and the final preprocessed data set. The experimental study, the results of which have been contrasted through nonparametric statistical tests, shows that our proposal obtains high reduction rates on training sets which greatly enhance the behavior of the nearest neighbor classifier.

1. Introduction

Classification is one of the best known tasks in machine learning [5,72]. Starting from an already processed training set, machine learning methods are able to extract knowledge from the data, which can be used to characterize new samples and classify them into classes already specified by the domain of the problem.

In recent years, there has been a manifold increase in the size of the data which these machine learning methods must manage [6]. Researchers in many application fields have developed more efficient and accurate data acquisition methods, which have allowed them to face greater and more difficult problems than before [45]. Therefore, the amount of data extracted to analyze those new challenges has grown to a point at which many classical data mining methods do not work properly, or, at least, suffer several drawbacks in their application.

Data reduction [54] is a data preprocessing task which can be applied to ease the problem of dealing with large amounts of data. Its main objective is to reduce the original data by selecting the most representative information. In this way, it is possible to avoid excessive storage and time complexity, improving the results obtained by any data mining application. The best known data reduction processes are feature selection (FS) [43], feature generation [28], attribute discretization [39], instance generation [65,66] and instance selection (IS) [21,41,42].

* Corresponding author. Tel.: +34 958 240598; fax: +34 958 243317.
E-mail addresses: jderrac@decsai.ugr.es (J. Derrac), Chris.Cornelis@UGent.be (C. Cornelis), sglopez@ujaen.es (S. García), herrera@decsai.ugr.es (F. Herrera).

0020-0255/$ - see front matter © 2011 Elsevier Inc. All rights reserved.
doi:10.1016/j.ins.2011.09.027

The K-Nearest Neighbors classifier (K-NN) [13,48,58] can be greatly enhanced when using these data reduction techniques. It is a nonparametric classifier which simply uses the entire input data set to establish the classification rule. Thus, the effectiveness of the classification process performed by the K-NN classifier relies mainly on the quality of the training data. Also, it is important to note that its main drawback is its relative inefficiency as the size of the problem increases, regarding both the number of examples in the data set and the number of attributes which will be used in the computation of the similarity function (distance) [11,33,68]. The K-NN classifier is one of the most relevant algorithms in data mining [73], being the best known Lazy Learning [1] method.

Recently, rough set theory (RST) [50,53] has been used to tackle several data mining problems with success. Focusing on the FS problem, RST can be used as a tool to extract a minimal set of features from the original data set (*decision reducts*), preserving the underlying semantics of the data while allowing reasonable generalization capabilities for the classifier [10,74]. This approach can be enhanced in several ways; for example, tolerance-based rough sets [60] provide an advanced way of defining approximation spaces and related similarity measures [46,61,62].

In addition, fuzzy logic [75] can also be hybridized with RST, obtaining as a result fuzzy rough feature selection methods, which offer greater flexibility and better potential to produce good-sized, high-quality feature subsets than the crisp ones [12,30,34,35,67]. Another key trait of fuzzy rough feature selection methods is that they can be applied directly over data sets representing continuous data, in contrast with pure RST feature reducers, which cannot be applied over continuous data sets without discretizing them at a previous step. Although discretization has proven to be a good technique for solving this kind of issues [39], in the K-NN classifier the use of continuous values is preferred due to the intrinsic characteristics of its decision rule (in fact, much research has been carried out in the opposite direction, turning discrete-based similarities values for the K-NN classifier into continuous ones [70]).

Evolutionary algorithms (EAs) [17] are general-purpose search algorithms that use principles inspired by nature to evolve solutions to problems. They have been successfully applied in different data mining problems [19,25,49]. Given that IS and FS tasks can be defined as combinatorial problems, it is possible to carry them out by using EAs [15]. In fact, many successful evolutionary proposals (most of them based on Genetic Algorithms (GAs)) have been developed to tackle them [2,7,20,29,31,32,38,47]. In particular, our previous work [14] proposes an evolutionary method for dealing with both IS and FS tasks simultaneously, using a ensemble based on preprocessed training sets.

In this work we present a new hybrid approach considering both fuzzy RST based FS and evolutionary IS, which we denote as EIS-RFS (evolutionary instance selection enhanced by Rough set based feature selection). A steady-state GA is developed to conduct the search for a suitable subset of instances, whereas the most useful features are selected by an heuristic-based fuzzy RST method. In this way, proper feature subsets can be selected during the search, taking advantage of using the information about indiscernibility already present in the training set.

Moreover, these subsets are considered within the framework of the GA, thus modifying the environment in which the instances are chosen. At the end of its application, EIS-RFS reports the best subsets found, which can be used to construct a reduced version of the training set, well suited to be used as a reference set for the K-NN classifier.

The performance of EIS-RFS is studied, comparing it with the K-NN classifier over unreduced data, considering different numbers of neighbors. Moreover, we test our approach further introducing a comparative study with several related techniques for IS and FS, considering a large set of standard classification problems. Finally, we also test its performance over large data sets, with a higher number of instances and features. All the results obtained have been contrasted using nonparametric statistical techniques [14], reinforcing the conclusions obtained.

The rest of the paper is organized as follows. In Section 2, some background about evolutionary IS and fuzzy RST based FS is given. In Section 3, the main characteristics of EIS-RFS are explained. Section 4 presents the experimental framework. Section 5 shows the achieved results. Section 6 summarizes our conclusions. Finally, Appendix A extends the description of the contrast estimation, one of the nonparametric tests employed in the study.

2. Background

This section covers the background information necessary to define and describe our proposal. It focuses on two topics: IS and FS as data reduction techniques (Section 2.1), and the use of fuzzy RST for FS (Section 2.2).

2.1. Instance and feature selection

IS is one of the main data reduction techniques. In IS, the goal is to isolate the smallest set of instances which enable a data mining algorithm to predict the class of a query instance with the same quality as the initial data set [41]. By minimizing the data set size, it is possible to reduce the space complexity and decrease the computational cost of the data mining algorithms that will be applied later, improving their generalization capabilities through the elimination of noise.

More specifically, IS can be defined as follows: Let $(\mathcal{X}, \mathcal{A})$ be an information system, where $\mathcal{X} = \{x_1, \ldots, x_n\}$ and $\mathcal{A} = \{a_1, \ldots, a_m\}$ are finite, non-empty sets of instances and features. Then, let us assume that there is a training set TR which consists of N instances and M features ($M = |\mathcal{A}|$), and a test set TS composed of T instances ($TR \cup TS = (\mathcal{X}, \mathcal{A})$). Let $S \subseteq TR$ be the subset of selected samples that resulted from the execution of an IS algorithm, then we classify a new pattern T from TS by a data mining algorithm acting over the instances of S.

IS methods can be divided into two categories: Prototype Selection (PS) methods and Training Set Selection (TSS) methods. PS methods [21] are IS methods which expect to find training sets offering the best classification accuracy and reduction rates by using instance based classifiers which consider a certain similarity or distance measure (for example, K-NN). On the other hand, TSS methods are known as the application of IS methods over the training set to build any predictive model (e.g. decision trees, subgroup discovery, neural networks [8,9,37]). In this work, we will focus on PS, since the nearest neighbor Rule will be used as the baseline rule to perform the classification process.

A key property of PS methods is given in [71]. There, Wilson and Martinez suggest that the determination of the k value in the K-NN classifier may depend on the proposal of the IS algorithm. Setting k as greater than 1 decreases the sensitivity of the algorithm to noise and tends to smooth the decision boundaries. In some IS algorithms, a value $k > 1$ may be convenient, when the interest lies in protecting the classification task against noisy instances. Therefore, they state that it may be appropriate to find a value of k to use during the reduction process, and then redetermine the best value of k in the classification task. In this study, we will test the differences found when using several values for k in EIS-RFS, although we recommend using the value $k = 1$, given that our EA needs to have the greatest possible sensitivity to noise during the reduction process. In this way, an evolutionary IS algorithm can better detect the noisy and redundant instances in order to find a subset of instances adapted to the simplest method of nearest neighbors.

Many approaches to PS have been developed [21]. Concerning evolutionary IS [15], the first contribution was made by Kuncheva et al. [38]. Interest in this field was increased by the study performed by Cano et al. [7], where a complete study of the use of EAs in IS was made. They concluded that EAs outperform classical algorithms both in reduction rates and classification accuracy. Therefore, research in this field has grown recently, with a wide number of noteworthy proposals [2,14,20,24,26,29].

FS is another of the main data reduction techniques. In FS, the goal is to select the most appropriate subset of features from the initial data set. It aims to eliminate irrelevant and/or redundant features to obtain a simple and accurate classification system [43].

Starting from the definition given for IS, FS can be defined as follows: Let us assume $\mathcal{A}, \mathcal{X}, TR$ and TS have already been defined. Let $B \subseteq \mathcal{A}$ be the subset of selected features that resulted from the execution of an FS algorithm over TR, then we classify a new pattern from TS by a data mining algorithm acting over TR, employing as a reference only the features selected in B.

There are three main categories into which FS methods can be classified:

- *Wrapper methods*, where the selection criterion is dependent on the learning algorithm, being a part of the fitness function [55].
- *Filtering methods*, where the selection criterion is independent of the learning algorithm (separability measures are employed to guide the selection) [27].
- *Embedded methods*, where the search for an optimal subset is built into the classifier construction [57].

As with IS methods, a large number of FS methods have been developed recently. Two of the most well known classical algorithms are forward sequential and backward sequential selection [40], which begin with a feature subset and sequentially add or remove features until the finalization of the algorithm. Some complete surveys, analyzing both classical and advanced approaches to FS, can be found in the literature [27,57,44].

2.2. Fuzzy RST for FS

In rough set analysis [51,52], each attribute a in $\mathcal{A}$ corresponds to an $\mathcal{X} \to V_a$ mapping, in which V_a is the value of a over $\mathcal{X}$. For every subset B of $\mathcal{A}$, the B-indiscernibility relation R_B is

$$R_B = \{(x,y) \in \mathcal{X}^2 \quad \text{and} \quad (\forall a \in B)(a(x) = a(y))\} \tag{1}$$

Therefore, R_B is an equivalence relation. Its equivalence classes $[x]_{R_B}$ can be used to approximate concepts, that is, subsets of the universe $\mathcal{X}$. Given $A \subseteq \mathcal{X}$, its lower and upper approximation with respect to R_B are defined by

$$R_B \downarrow A = \{x \in \mathcal{X} | [x]_{R_B} \subseteq A\} \tag{2}$$

$$R_B \uparrow A = \{x \in \mathcal{X} | [x]_{R_B} \cap A \neq \emptyset\} \tag{3}$$

A *decision system* $(\mathcal{X}, \mathcal{A} \cup \{d\})$ is a special kind of information system, used in the context of classification, in which $d (d \notin \mathcal{A})$ is a designated attribute called the decision attribute. Its equivalence classes $[x]_{R_d}$ are called decision classes. Given $B \subseteq A$, the B-positive region POS_B contains those objects from X for which the values of B allow to predict the decision class unequivocally:

$$POS_B = \bigcup_{x \in X} R_B \downarrow [x]_{R_d} \tag{4}$$

Indeed, if $x \in POS_B$, it means that whenever an instance has the same values as x for the attributes in B, it will also belong to the same decision class as x. The predictive ability with respect to d of the attributes in B is then measured by the following value (degree of dependency of d on B):

$$\gamma_B = \frac{|POS_B|}{|\mathcal{X}|} \tag{5}$$

A subset B of $\mathcal{A}$ is called a decision reduct if it satisfies $POS_B = POS_{\mathcal{A}}$, that is, B preserves the decision making power of $\mathcal{A}$, and moreover it cannot be further reduced; in other words, there is no proper subset B' of B so that $POS_{B'} = POS_{\mathcal{A}}$. If the latter constraint is lifted – B is not necessarily minimal – we call B a decision superreduct.

Instead of using a crisp equivalence relation R to represent objects' indiscernibility, we can also measure their approximate equality by means of a fuzzy relation R. Typically, we assume that R is at least a fuzzy tolerance relation; in other words, R is reflexive and symmetric.

Assuming that for a qualitative attribute a, the classical way of discerning objects is used, that is, $R_a(x,y) = 1$ if $a(x) = a(y)$ and $R_a(x,y) = 0$ otherwise, we can define, for any subset B of $\mathcal{A}$, the fuzzy B-indiscernibility relation by

$$R_B(x,y) = \mathcal{T}(\underbrace{R_a(x,y)}_{a\in B}) \tag{6}$$

in which $\mathcal{T}$ represents a t-norm. It can be seen that if only qualitative attributes (possibly originating from discretization) are used, then the traditional concept of B-indiscernibility relation is recovered.

For the lower approximation of a fuzzy set A in X by means of a fuzzy tolerance relation R, we adopt the definitions of [56]: given an implicator $\mathcal{I}$ and a t-norm $\mathcal{T}$, formulas (2) and (3) were paraphrased to define $R{\downarrow}A$ and $R{\uparrow}A$ by

$$(R \downarrow A)(y) = \inf_{x\in X} \mathcal{I}(R(x,y), \quad A(x)), \quad \forall y \in X \tag{7}$$

$$(R \uparrow A)(y) = \sup_{x\in X} \mathcal{T}(R(x,y), \quad A(x)), \quad \forall y \in X \tag{8}$$

Using fuzzy B-indiscernibility relations, we can define the fuzzy B-positive region by, for y in U,

$$POS_B(y) = \left(\bigcup_{x\in X} R_B \downarrow [\mathcal{X}_{R_d}]\right)(y) \tag{9}$$

This means that the fuzzy positive region is a fuzzy set in X, to which an object y belongs to the extent that its R_B-foreset is included into at least one of the decision classes.

While formula (9) provides the most faithful way to define the fuzzy positive region, it was shown in [12] that

$$POS_B(y) = (R_B \downarrow R_d y)(y) \tag{10}$$

becomes Eq. (9) when the decision feature is crisp.

Once we have fixed the fuzzy positive region, we can define an increasing [0,1]-valued measure to gauge the degree of dependency of a subset of features on another subset of features. For FS it is useful to phrase this in terms of the dependency of the decision feature on a subset of the conditional features:

$$\gamma_B = \frac{|POS_B|}{|POS_{\mathcal{A}}|} \tag{11}$$

3. An evolutionary fuzzy RST based model for feature and instance selection: EIS-RFS

This section is devoted to analyzing and describing EIS-RFS, and its main components, from a bottom-up perspective. Therefore, the first step will be to describe the GA employed to conduct the search of the subsets of instances (Section 3.1). These subsets will be optimized with the inclusion of a fuzzy RST-based FS procedure (Section 3.2). Finally, the EIS-RFS framework will be defined as a cooperation between the former procedures (Section 3.3).

3.1. A steady-state GA for IS

The IS component of EIS-RFS is guided by an evolutionary method. Specifically, we have opted to develop a steady-state GA to accomplish this task.

Steady-state GAs are GAs in which only a reduced (and fixed) set of offspring are produced in each generation (usually one or two).Parents are selected to produce offspring and then a decision is made as to which individuals in the population will be selected for deletion in order to make room for the new offspring.

In the development of the steady-state GA for our approach, the following choices have been made:

- *Codification*: The steady-state GA will use binary chromosomes to represent the solutions. Each bit will represent the state of each instance in the training set (**1** if the instance is selected; **0** if it is deleted), as is usually done in Evolutionary IS approaches [15].

- *Selection of parents*: Binary tournament procedure will be used to select parents in each generation.
- *Crossover operator*: A two-point crossover operator has been considered. In each generation, this operator is applied twice, obtaining as a result two offspring.
- *Mutation operator*: The bit-flip mutation operator (changing the value of the selected allele from 0 to 1, and vice versa) is applied to each offspring produced, with a given probability per bit.
- *Replacement strategy*: The two worst individuals of the population are chosen for replacement, only if their fitness value is lower than the offspring's.

Algorithm 1 shows a basic pseudocode of the steady-state GA.

Algorithm 1. Steady-state GA algorithm basic structure

```
Input:A population
Output:An optimized population
Initialize population;
    while Termination criterion not satisfied do
    Select two parents from the population;
    Create two offspring using crossover and mutation;
    Evaluate the offspring with the Fitness function;
    Select two individuals in the population, which may be replaced by the offspring;
    Decide if this/these individuals will be replaced;
end
```

Concerning the fitness function, it must pursue both reduction and accuracy objectives when evaluating an IS chromosome, J. To do so, we will follow the proposal given in [7], where Cano et al. defined *AccRate* as the accuracy achieved by a K-NN classifier when classifying the entire training set using the currently selected subset as a reference and using leave-one-out as validation scheme

$$AccRate(J) = \mathrm{K-NNAccuracy}(J) \tag{12}$$

RedRate as the reduction rate achieved over the currently selected (maintained) instances

$$RedRate(J) = \frac{\#\mathrm{Instances\ Selected(J)}}{N} \tag{13}$$

and a real-valued weighting factor, α, to adjust the strength of each term in the resulting fitness value. Eq. (14) defines the full fitness function

$$Fitness(J) = \alpha \cdot AccRate(J) + (1-\alpha) \cdot RedRate(J) \tag{14}$$

Following the recommendations given in [7], EIS-RFS will employ a value $\alpha = 0.5$, which should offer an adequate trade-off between accuracy and reduction goals.

3.2. Selecting features by means of a fuzzy RST procedure

The concept of discernibility, defined in the realm of fuzzy RST, allows several useful approaches to data reduction to be developed [12,34]. The elements of a given data set can be analyzed, identifying which are discernible (with respect to the elements belonging to the other classes of the domain), regarding the specific set of features considered.

Therefore, a straightforward method to properly characterize a specific training set is to select those features which are able to fully discern all the instances of the training set (or, at least, discern them as much as possible). This way, the pruned training set can maintain its capabilities of separating instances belonging to different classes (or even increase them, by the removal of noisy and irrelevant features), while its size is reduced.

Eq. (11) gives a proper measure to evaluate the discernibility of a subset of features. The first step to compute this measure consists of defining a similarity measure between two different values of a same feature. This measure can be modeled as a fuzzy tolerance relation, R. For quantitative values, a suitable measure was defined in [36]:

$$R_a(x,y) = max\left(min\left(\left(\frac{a(y)-a(x)+\sigma_a}{\sigma_a}, \frac{a(x)-a(y)+\sigma_a}{\sigma_a}\right), 0\right)\right) \tag{15}$$

where x and y are two different instances belonging to the training set, and σ_a denotes the standard deviation of a. For nominal attributes, instead of using the equality metric:

$$R_a(x,y) = \begin{cases} 1 & \text{if } a(x) = a(y) \\ 0 & \text{otherwise} \end{cases} \tag{16}$$

we propose the Value Difference Metric (VDM) [63,70], where two values are considered to be closer if they have more similar classifications (that is, more similar correlations with the output classes).

$$R_a(x,y) = \sum_{c=1}^{C} \left| \frac{N_{a,x,c}}{N_{a,x}} - \frac{N_{a,y,c}}{N_{a,y}} \right|^q \tag{17}$$

where:

- $N_{a,x}$ is the number of instances in the training set that have the same value as instance x for attribute a.
- $N_{a,x,c}$ is the number of instances in the training set that have the same value as instance x for attribute a and output class c.
- C is the number of output classes in the problem domain.
- q is a constant (typically 2).

Although the joint use of the overlap metric with crisp connectives would lead to an approach similar to those based on classical RST, the use of VDM is preferred, since it provides a soft similarity measure, suitable to be combined with the use of fuzzy connectives.

Eqs. (16) and (17) allows us to employ the fuzzy B-indiscernibility relation defined by Eq. (6), in which $\mathcal{T}$ represents the minimum t-norm, $\mathcal{T}(x,y) = min(x,y), x,y \in [0,1]$. Then, it is possible to compute the lower approximation of a fuzzy set A in $\mathcal{X}$ by means of a fuzzy tolerance relation R, by using Eq. (7) (employing, as implicator, $\mathcal{I}$, the Lukasiewicz one, $\mathcal{I}(x,y) = min(1, 1-x+y), x,y \in [0,1]$). With these definitions, it is possible to obtain the degree of inclusion of the instances of the training set in the fuzzy B-positive region, POS_B, and the gamma measure for B, γ_B.

Once a suitable measure of quality for a given subset of features, B, has been defined (γ_B), a search procedure to find the best possible subset can be carried out. Following the recommendations given in [12], the QUICKREDUCT heuristic [35] will be employed. Algorithm 2 shows its basic pseudocode.

Algorithm 2. QUICKREDUCT heuristic

```
Input:A set of instances
Output:A subset of features (B)
B ← {};
repeat
    T ← B,best ←-1;
    foreach a∈ (A \ B) do
      If γ_{B∪{a}}> best then
        T ← B ∪ {a}, best ←γ_{B∪{a}};
      end
    end
    B ← T;
until γ_B ⩾ MaxGamma;
```

Basically, QUICKREDUCT considers all the features in the domain of the problem and tries to add them to the candidate subset T. Features are added only if γ_B is improved. This hillclimbing procedure is continued until an established *MaxGamma* value (typically 1) is reached.

This filter method enables suitable subsets of features (with $\gamma_B = 1$) to be found quickly, which properly represent the information contained in the data set considered as the input parameter. Note that, when this method is used inside the framework of a larger algorithm, either subsets of instances or the entire training set can be considered.

3.3. Hybrid model for simultaneous IS and FS

Once the two basic tools considered for performing IS and FS have been defined, we can describe the hybrid model which composes our approach. Basically, it is a steady-state GA for IS where, every time a fixed number of evaluations has been spent, an RST based FS procedure is applied to modify the features considered during the search. Therefore, at any time only a single feature subset will be used in the whole search procedure. As the search progresses, this subset will be upgraded and adapted, to fit with the best subset of instances found.

Fig. 1 shows a flowchart representing its main steps: Initialization (Step 1), feature selection procedure (Step 4), Instance Selection procedure (Step 5), and Output (Step 7). The rest of the operations (Steps 2,3 and 6) control whether each of the former procedures should be carried out. The properties of each step are detailed as follows:

1. **Initialization**: The initialization procedure consists of the initialization of the chromosomes of the population, and the selection of the initial subset of features. The chromosomes, representing different subsets of instances, are initialized randomly (taking binary values, that is, in {0,1}). Regarding the initial subset of features, two different subsets are considered:

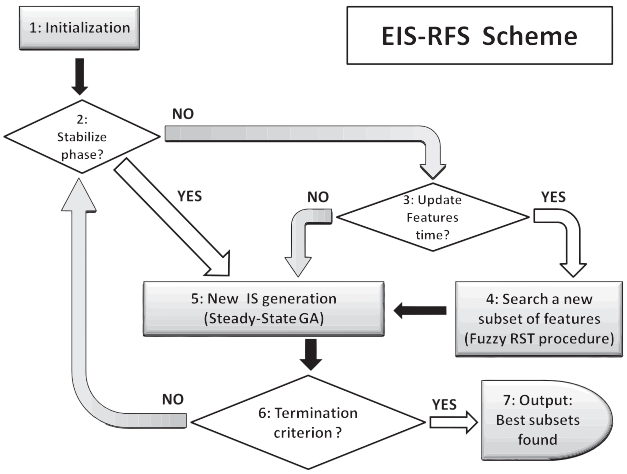

Fig. 1. Flowchart depicting the main steps of EIS-RFS. Rectangles depict processes whereas rhombuses depict decisions taken by the algorithm.

- The full set of features of the domain.
- The subset of features selected by the RST based FS method using the whole training set as input.

The best performing subset (that is, the one which achieves the lowest error when applied to a K-NN classifier) is selected as the global subset of features of EIS-RFS. Only the features included in this subset will be considered in subsequent phases, until new changes in the subset are made (step 4).

2. **Stabilize phase**: Changes in the current subset of features are only considered if the search is not near its end. Therefore, if the current number of evaluations spent is higher than $\beta \cdot$ *MAX_EVALUATIONS* (usually with β near to 1), the stabilize stage is activated and no further changes in the subset of features selected are considered. This mechanism allows EIS-RFS to easily converge for hard problems, where the final subset of features is fixed before the end of the search. It allows EIS-RFS to focus its last efforts on optimizing the subsets of instances selected, performing a final refinement of the solutions achieved.
3. **Update Features**: If the Stabilize phase is not activated yet, it checks whether the FS procedure will be started or not. It will be performed every time *UpdateFS* evaluations have been spent by the steady-state GA.
4. **Search for a new subset of features**: This procedure consists of using the RST-based FS filter method, using as input the current best chromosome of the population (the best subset of instances found so far). The new subset of features obtained is tested by applying it to a K-NN classifier (considering as a reference set only the current best subset of instances). If this subset performs better than the former, it is accepted as the global subset of features of EIS-RFS.
5. **New IS generation**: An IS generation is carried out using the steady-state GA scheme described in Section 3.1. Note that, when evaluating a new chromosome, the K-NN classifier used in the fitness function will only consider the selected features in the global subset of features of EIS-RFS.
6. **Termination criterion**: The search process of EIS-RFS ends if the number of evaluations spent reaches the *MAX_EVALUATIONS* limit. Otherwise, a new cycle of the algorithm begins.
7. **Output: Best subsets found**: When the fixed number of evaluations runs out, the best chromosome of the population is selected as the best subset of instances found. The current global subset of selected features is designed as the best subset of features found. Both subsets are returned as the output of EIS-RFS.

The output of EIS-RFS (a subset of instances and a subset of features) defines a pruned version of the original training set. This set is ready to be used as a reference set by a K-NN classifier to perform a faster and more accurate classification of new test instances.

4. Experimental framework

This section presents the experimental study designed to test our proposal. Section 4.1 presents the classification data sets used throughout the study. Section 4.2 summarizes the algorithms selected for the comparison and the statistical procedures applied.

4.1. Data sets

To check the performance of EIS-RFS, we have selected a set of 43 classification data sets (30 standard data sets and 13 large data sets). These are well-known problems in the area, taken from the UCI Machine Learning Repository [18] and the

Table 1
Summary description for standard data sets.

Data Set	#Ex.	#Feat.	#Num.	#Nom.	#Cl.	Data set	#Ex.	#Feat.	#Num.	#Nom.	#Cl.
Australian	690	14	8	6	2	Iris	150	4	4	0	3
Balance	625	4	4	0	3	Led7digit	500	7	7	0	10
Bupa	345	6	6	0	2	Lymphography	148	18	3	15	4
Cleveland	303	13	13	0	5	Mammographic	961	5	5	0	2
Contraceptive	1,473	9	9	0	3	Monk-2	432	6	6	0	2
Crx	690	15	6	9	2	Newthyroid	215	5	5	0	3
Ecoli	336	7	7	0	8	Pima	768	8	8	0	2
Flare-solar	1,066	11	0	11	6	Saheart	462	9	8	1	2
German	1,000	20	7	13	2	Sonar	208	60	60	0	2
Glass	214	9	9	0	7	Spectheart	267	44	44	0	2
Haberman	306	3	3	0	2	Tic-tac-toe	958	9	0	9	2
Hayes-roth	160	4	4	0	3	Wine	178	13	13	0	3
Heart	270	13	13	0	2	Wisconsin	699	9	9	0	2
Hepatitis	155	19	19	0	2	Yeast	1,484	8	8	0	10
Housevotes	435	16	0	16	2	Zoo	101	16	0	16	7

Table 2
Summary description for large data sets.

Data Set	#Ex.	#Feat.	#Num.	#Nom.	#Cl.	Data Set	#Ex.	#Feat.	#Num.	#Nom.	#Cl.
Abalone	4,174	8	7	1	28	Satimage	6,435	36	36	0	7
Banana	5,300	2	2	0	2	Segment	2,310	19	19	0	7
Chess	3,196	36	0	36	2	Spambase	4,597	57	57	0	2
Marketing	8,993	13	13	0	9	Splice	3,190	60	0	60	3
Mushroom	8,124	22	0	22	2	Titanic	2,201	3	3	0	2
Page-blocks	5,472	10	10	0	5	Twonorm	7,400	20	20	0	2
Ring	7,400	20	20	0	2						

KEEL-dataset repository[1] [3,4]. Tables 1 and 2 summarizes their properties. For each data set, we provide its number of examples (#Ex.), Features (#Feat.) the number of numerical (#Num.) and nominal (#Nom.) attributes, and the number of classes (#Cl.). For continuous attributes, their values are normalized in the interval [0,1] to equalize the influence of attributes with different range domains.

The data sets considered are partitioned by using the ten fold cross-validation (10-fcv) procedure [64], that is, each data set is randomly partitioned into ten subsets, preserving the same size (the same number of instances) and the same class distribution between partitions. In an iterative process, one partition is selected as the test set whereas the training set is composed of the rest. Final results are obtained averaging the results obtained over the ten partitions (stochastic algorithms have been run three times).

4.2. Algorithms, parameters and statistical analysis

Several evolutionary preprocessing methods for the K-NN classifier have been selected to perform a comprehensive study of the capabilities of our approach. These methods are the following:

- **IS-SSGA**: A steady-state GA for IS. This evolutionary method has the same characteristics as EIS-RFS, but does not include any kind of feature selection process. Its behavior as an IS method was studied in depth in [7].
- **FS-SSGA**: A steady-state GA for FS. The features of the domain are encoded in the binary chromosomes, in a similar way to IS-SSGA. Note that in the fitness function of this method, the reduction rate is computed over the ratio of features selected. Therefore, its α weight must be very near to 1.0, to avoid an excessive deletion of features which may degrade the accuracy of the classifier too much.
- **IFS-SSGA**: A steady-state GA for simultaneous IS and FS. Its binary coded chromosomes encode both instances and features of the domain. The fitness function of this method only considers instances' reduction rate to compute the reduction ratio achieved.
- **FS-RST**: The fuzzy RST based feature selection method (FS-RST) used within the EIS-RFS framework, applied in isolation.
- **FS-RST + IS-SSGA**: FS-RST used as a preprocessor to IS-SSGA, that is, the preprocessed reference set obtained after the application of FS-RST is used as the input of the IS-SSGA method.
- **IS-SSGA + FS-RST**: IS-SSGA used as a preprocessor to FS-RST, that is, the preprocessed reference set obtained after the application of IS-SSGA is used as the input of the IS-SSGA method.

[1] http://www.keel.es/datasets.php.

Table 3
Parameter specification for the algorithms tested in the experimentation.

Algorithm	Parameters (all K-NN based methods will use $k = 1$, unless explicitly stated in a particular experiment)
EIS-RFS	MAX_EVALUATIONS: 10000, Population size: 50, Crossover probability: 1.0, Mutation probability: 0.005 per bit, α: 0.5 *MaxGamma*: 1.0, UpdateFS: 100, β: 0.75
IS-SSGA	MAX_EVALUATIONS: 10000, Population size: 50, Crossover probability: 1.0, Mutation probability: 0.005 per bit, α: 0.5
FS-SSGA	MAX_EVALUATIONS: 10000, Population size: 50, Crossover probability: 1.0, Mutation probability: 0.005 per bit, α: 0.99
IFS-SSGA	MAX_EVALUATIONS: 10000, Population size: 50, Crossover probability: 1.0, Mutation probability: 0.005 per bit, α: 0.5
FS-RST	*MaxGamma*: 1.0

In order to estimate the classification accuracy, the preprocessed training sets obtained as a result of the application of these methods are tested by using a K-NN classifier.

Many different configurations can be established for each combination of domain and method. However, for the sake of a fair comparison, we have selected a fixed set of parameters for each method. Table 3 summarizes them.

Hypothesis testing techniques are used to contrast the experimental results and provide statistical support for the analysis [59]. Specifically, we use non-parametric tests, since the initial conditions that guarantee the reliability of the parametric tests may not be satisfied, causing the statistical analysis to lose credibility [16,23].

Throughout the study, two nonparametric tests for pairwise statistical comparisons of classifiers will be employed. The first one is the well-known Wilcoxon Signed-Ranks Test [69]. The second one is the Contrast Estimation of medians [22], which is very useful for estimating the difference between two algorithms' performance. We describe its detailed definition in Appendix A.

Further information about these tests and other statistical procedures specifically designed for use in the field of Machine Learning can be found at the SCI2S thematic public website on *Statistical Inference in Computational Intelligence and Data Mining*.[2]

5. Results and analysis

This section is devoted to presenting and analyzing the results achieved in several studies performed to test the behavior of EIS-RFS and compare it with several related techniques. To this end, Section 5.1 studies the behavior of EIS-RFS and K-NN as the number of neighbors selected grow. Section 5.2 shows a comparison between EIS-RFS and the rest of the comparison methods selected in standard data sets. Finally, Section 5.3 compares EIS-RFS when applied to large data sets.

5.1. EIS-RFS vs K-NN

One of the most critical issues of the K-NN classifier lies in the selection of its k parameter, that is, the number of neighbors considered in the decision rule. Usually, an odd number of them is preferred, since this way ties in the decision of the class membership are less likely to occur.

Concerning EIS-RFS, this is also a key issue, since the specific number of neighbors selected will affect the way in which the accuracy assigned to the chromosomes is estimated, thus modifying its behavior during the search.

To analyze this issue, we have classified the 30 standard data sets with EIS-RFS and K-NN, using 1, 3, 5 and 7 neighbors. Table 4 shows the average accuracy results achieved (the best results in each data set and category are highlighted in bold) and the number of times that each method achieves the best result. Table 5 presents the results of the Wilcoxon Signed-Ranks Test performed to contrast the results in each category.

The results show the validity of EIS-RFS as a preprocessing method for the K-NN classifier. Its average accuracy is improved in every category studied (from 1 to 7 neighbors), and the number of data sets in which EIS-RFS offers a best result is always greater. Moreover, the Wilcoxon Signed-Ranks Test confirms that differences between the methods are significant at the 0.05 significance level.

Fig. 2 depicts this comparison graphically. The dots symbolize the accuracy achieved in test phase by EIS-RFS and K-NN in a concrete data set (30 points are represented in each graph). A straight line splits the graph, exactly at the points where the accuracy measure of both classifiers is equal. Therefore, those points below (right) of the line represent data sets where EIS-RFS behaves better than K-NN, whereas those points above (left) of the line represent the opposite.

Clearly, EIS-RFS outperforms K-NN in every case, albeit the improvement achieved with the application of preprocessing diminishes as the number of neighbors considered is increased. Similarly to IS algorithms, setting a value of k greater than 1 for EIS-RFS decreases its sensitivity to noise, smoothing the decision boundaries [71]. Therefore, the enhancement obtained if the number of neighbors selected is high will be lower, although its application will still be beneficial if its results are compared with those obtained without preprocessing data (K-NN).

Finally, a second conclusion arrived at this study is that EIS-RFS behaves slightly better if only 1 neighbor is considered. Thus, we will fix the number of neighbors considered by it to 1 in the rest of the experimental study.

[2] http://sci2s.ugr.es/sicidm/.

Table 4
Average accuracy rates obtained by EIS-RFS and K-NN considering different numbers of neighbors.

Data set	1 neighbor		3 neighbors		5 neighbors		7 neighbors	
	EIS-RFS	K-NN	EIS-RFS	K-NN	EIS-RFS	K-NN	EIS-RFS	K-NN
Australian	**85.66**	81.45	**85.91**	84.78	**84.98**	84.78	**85.65**	84.78
Balance	**85.92**	79.04	**85.74**	83.37	**86.88**	86.24	87.83	**88.48**
Bupa	**65.72**	61.08	**64.91**	60.66	**64.37**	61.31	**63.75**	62.53
Cleveland	**55.16**	53.14	**56.42**	54.44	**56.47**	55.45	**56.81**	56.45
Contraceptive	**45.42**	42.77	**46.85**	44.95	**47.59**	46.85	**49.16**	48.27
Crx	**84.93**	79.57	**84.49**	84.20	85.07	**85.51**	**85.80**	85.65
Ecoli	**82.14**	80.70	79.84	**80.67**	80.55	**81.27**	80.87	**82.45**
Flare-solar	**66.32**	55.54	**65.48**	55.07	**65.95**	57.04	**66.04**	63.89
German	**70.80**	70.50	**70.90**	69.60	**74.10**	71.80	**74.30**	72.20
Glass	67.35	**73.61**	65.36	**70.11**	64.90	**66.85**	65.10	**66.83**
Haberman	**71.56**	66.97	**72.60**	70.58	**72.49**	66.95	**72.19**	69.90
Hayes-roth	**80.86**	35.70	**74.98**	24.82	**67.98**	23.95	**62.59**	26.86
Heart	**80.74**	77.04	**79.56**	77.41	78.89	**80.74**	**80.37**	79.26
Hepatitis	**82.58**	82.04	83.13	**83.88**	83.04	**85.21**	83.29	**83.88**
Housevotes	**94.48**	91.24	**95.84**	94.01	**94.00**	93.31	**93.32**	93.09
Iris	**96.00**	93.33	**94.00**	**94.00**	**96.00**	**96.00**	95.33	**96.00**
Led7digit	**73.20**	40.20	**74.60**	45.20	**70.40**	41.40	**70.20**	43.40
Lymphography	**77.15**	73.87	**77.44**	77.39	**82.65**	79.44	**82.25**	81.49
Mammographic	**80.65**	76.38	**81.27**	79.19	**81.56**	81.06	**81.48**	81.17
Monk-2	**100.00**	77.91	**97.55**	96.29	**97.07**	94.75	**100.00**	89.16
Newthyroid	96.77	**97.23**	**95.41**	95.37	**94.91**	93.98	**96.32**	92.58
Pima	**74.80**	70.33	**73.45**	72.93	**73.72**	73.06	**74.50**	72.93
Saheart	**68.82**	64.49	**68.67**	68.18	**68.64**	67.10	**68.46**	66.45
Sonar	80.76	**85.55**	78.83	**83.07**	78.60	**83.10**	78.48	**80.21**
Spectfheart	**76.82**	69.70	**74.99**	71.20	**78.33**	71.97	**77.86**	77.58
Tic-tac-toe	**78.29**	73.07	**78.01**	77.56	78.12	**83.30**	78.69	**82.88**
Wine	**97.19**	95.52	94.35	**95.49**	**96.26**	96.05	96.35	**96.63**
Wisconsin	**96.42**	95.57	**96.33**	96.00	96.28	**96.57**	96.14	**97.00**
Yeast	**53.37**	50.47	**56.20**	53.17	55.86	**56.74**	**57.55**	57.49
Zoo	**96.39**	92.81	**94.81**	92.81	**94.97**	93.64	**94.64**	92.97
Average	**78.88**	72.89	**78.26**	74.55	**78.35**	75.18	**78.51**	75.75
Best result (of 30)	27	3	25	6	21	10	21	9

Table 5
Wilcoxon Signed-Ranks Test results for EIS-RFS vs K-NN.

EIS-RFS vs K-NN	R^+	R^-	P-value
1 neighbor	418.0	47.0	0.00004
3 neighbors	357.0	78.0	0.00182
5 neighbors	310.0	125.0	0.04552
7 neighbors	335.5	129.5	0.03363

5.2. Comparison with IS, FS and hybrid techniques

Table 6 shows the results measured by accuracy in test data for each method considered in the study. For each data set, the mean accuracy and the standard deviation are computed. The best case in each data set is highlighted in bold. The last row in each table shows the average considering all data sets.

On the other hand, Table 7 shows the average reduction rates achieved. In instances' search space, the reduction is defined by Eq. (13). The reduction rate for features is computed in a similar way, considering only those features selected (maintained) in the final reference set

$$RedRate_{\text{Features}} = \frac{\#\text{Features Selected}}{M} \tag{18}$$

In each category, only those methods that perform any kind of reduction are considered, that is:

- Instances' search space: EIS-RFS, IS-SSGA, IFS-SSGA, FS-RST + IS-SSGA, IS-SSGA + FS-RST.
- Features' search space: EIS-RFS, FS-SSGA, IFS-SSGA, FS-RST, FS-RST + IS-SSGA, IS-SSGA + FS-RST.

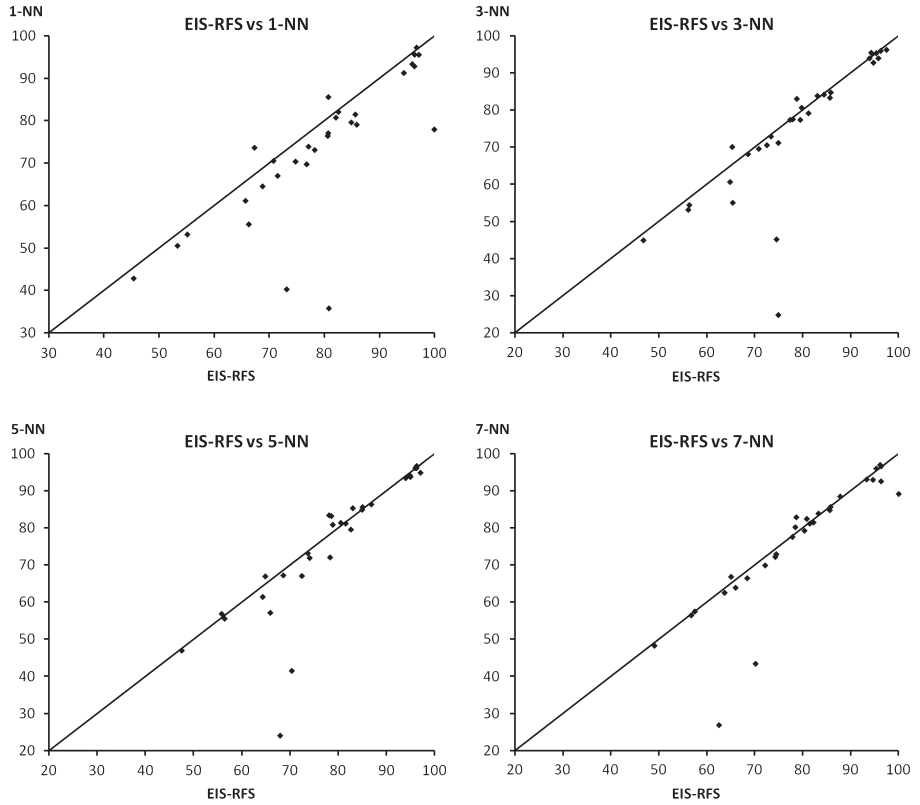

Fig. 2. Graphical comparison of EIS-RFS and K-NN using 1, 3, 5 and 7 neighbors. EIS-RFS shows a better performance in every case. However this improvement is lower as the number of neighbors increases.

Finally, Table 8 reports the time elapsed by each method in training phase (in seconds).[3] Note that running times in test phase are not reported due to the fact that they are too low to show interesting differences and efficiency in test phase is already reflected by the reduction rates achieved (the higher the reduction rates are, the less running time will be needed).

Reading the results shown in the tables, we can make the following analysis:

- EIS-RFS achieves the best accuracy result, obtaining the best overall result in 13 of the 30 data sets.
- Concerning reduction in instances' space, all the methods considered achieve a similar rate. Hence, the computational time spent by the K-NN classifier in test phase will be very low if any of the reference sets produced is used, including the one preprocessed by EIS-RFS.
- In the features' space, EIS-RFS shows a similar behavior to FS-RST. Although these reduction rates are low when compared with the ones achieved by the evolutionary techniques (FS-SSGA and IFS-SSGA), these methods could delete too much features due to the fitness function used. Note that, within the evolutionary search (and especially in its first steps), it is easier to remove features than to find an accurate combination of them in order to improve the fitness value. Thus, by sacrificing some of its reduction power, EIS-RFS is able to find better subsets of features, reinforcing our hypothesis of hybridizing Fuzzy-RST with EIS.
- Concerning the time elapsed, EIS-RFS spent a slightly higher time than the rest of methods, except for FS-RST (whose computational time is not comparable since it only evaluates a single reference set, instead of the 10,000 evaluations performed by the evolutionary methods), and FS-SSGA, whose time requirements are twice those of the rest.

Regarding the statistical analysis performed, Table 9 presents the results of the Wilcoxon Signed Ranks test, while Table 10 shows the results of the Contrast Estimation. Note that these tests have been carried out considering the average accuracy of the results obtained.

The Wilcoxon test shows that our approach, EIS-RFS, statistically outperforms all the comparison methods with a level of significance $\alpha = 0.01$; that is, no comparison of EIS-RFS and any comparison method achieves a P-value equal to or higher than 0.01. This is a strong result, which supports the fact that EIS-RFS clearly outperforms all the other techniques.

[3] The experiments have been carried out on a machine with a Dual Core 3,20 GHz processor and 2 GB of RAM, running under the Fedora 4 operating System.

Table 6
Accuracy results in test phase.

Data set	EIS-RFS	IS-SSGA	FS-SSGA	IFS-SSGA	FS-RST	FS-RST+IS-SSGA	IS-SSGA+ FS-RST
Australian	**85.66** ± 2.27	85.65 ± 2.77	85.07 ± 3.49	85.36 ± 3.31	81.45 ± 4.52	85.51 ± 2.92	80.87 ± 4.49
Balance	85.92 ± 2.62	**86.40** ± 3.08	70.89 ± 9.80	84.31 ± 4.85	79.04 ± 6.81	71.89 ± 3.28	61.05 ± 6.82
Bupa	**65.72** ± 8.79	61.14 ± 9.37	59.91 ± 10.19	62.72 ± 8.40	62.51 ± 7.78	61.14 ± 9.60	61.14 ± 7.17
Cleveland	55.16 ± 5.82	52.82 ± 4.47	51.47 ± 9.47	**56.13** ± 6.07	52.51 ± 9.49	54.42 ± 5.07	51.51 ± 9.37
Contraceptive	**45.42** ± 5.14	44.54 ± 4.61	41.96 ± 3.57	45.15 ± 2.32	42.63 ± 3.73	44.54 ± 4.92	45.15 ± 4.07
Crx	**84.93** ± 5.72	84.64 ± 4.22	81.16 ± 7.61	84.64 ± 5.08	81.30 ± 6.28	83.19 ± 4.79	80.72 ± 5.82
Ecoli	**82.14** ± 8.42	80.38 ± 5.69	78.90 ± 7.30	77.70 ± 5.52	76.58 ± 14.73	77.18 ± 6.49	75.97 ± 15.00
Flare-solar	66.32 ± 2.94	64.82 ± 3.37	62.76 ± 3.65	**67.35** ± 4.12	63.23 ± 5.56	64.82 ± 3.53	63.04 ± 5.31
German	**70.80** ± 4.24	70.40 ± 3.24	69.50 ± 2.68	70.10 ± 3.48	67.90 ± 3.41	69.20 ± 3.34	69.60 ± 3.20
Glass	67.35 ± 11.83	67.10 ± 14.74	71.80 ± 14.30	71.23 ± 10.64	**74.50** ± 13.17	67.10 ± 15.68	66.12 ± 13.66
Haberman	71.56 ± 7.34	71.23 ± 5.40	72.81 ± 6.15	**72.83** ± 5.99	65.68 ± 6.58	71.23 ± 6.20	71.23 ± 6.42
Hayes–roth	80.86 ± 11.70	69.15 ± 11.69	**83.93** ± 8.33	79.80 ± 11.65	76.07 ± 14.07	74.87 ± 11.13	77.95 ± 12.76
Heart	80.74 ± 6.34	81.11 ± 7.90	76.67 ± 6.06	**82.59** ± 6.31	78.89 ± 6.77	79.26 ± 8.84	71.48 ± 6.93
Hepatitis	**82.58** ± 7.99	79.33 ± 8.71	76.21 ± 7.89	80.67 ± 6.13	79.50 ± 7.95	80.04 ± 8.38	71.13 ± 7.52
Housevotes	**94.48** ± 3.67	93.79 ± 3.43	94.01 ± 4.53	94.46 ± 4.37	90.78 ± 6.47	74.42 ± 4.04	62.56 ± 5.99
Iris	**96.00** ± 4.92	94.67 ± 2.81	95.33 ± 4.50	94.67 ± 4.22	93.33 ± 5.44	94.67 ± 3.32	94.67 ± 5.73
Led7digit	73.20 ± 4.99	**73.40** ± 2.84	63.00 ± 6.94	71.40 ± 4.81	63.60 ± 5.87	17.40 ± 2.99	17.00 ± 6.15
Lymphography	77.15 ± 12.15	77.92 ± 9.39	**78.49** ± 9.12	74.92 ± 10.79	77.38 ± 11.21	78.06 ± 9.87	66.55 ± 12.18
Mammographic	**80.65** ± 4.51	79.50 ± 3.85	75.86 ± 6.07	80.15 ± 6.23	75.76 ± 4.97	79.50 ± 3.85	79.50 ± 5.08
Monk-2	**100.00** ± 0.00	83.53 ± 6.21	**100.00** ± 0.00	98.64 ± 3.07	77.91 ± 5.71	**100.00** ± 5.95	96.13 ± 5.50
Newthyroid	96.77 ± 4.83	98.16 ± 3.20	96.30 ± 1.95	96.32 ± 3.60	97.23 ± 2.39	98.16 ± 3.58	**98.61** ± 2.23
Pima	**74.80** ± 3.71	72.26 ± 4.44	67.70 ± 4.59	73.83 ± 3.15	70.33 ± 3.71	72.26 ± 4.48	72.26 ± 3.67
Saheart	68.82 ± 7.16	**69.27** ± 3.70	61.24 ± 3.91	67.99 ± 5.69	64.49 ± 4.21	65.39 ± 4.09	66.04 ± 4.61
Sonar	80.76 ± 7.88	75.45 ± 11.74	**84.62** ± 8.65	75.50 ± 12.59	81.69 ± 9.83	71.10 ± 11.61	69.21 ± 9.62
Spectfheart	76.82 ± 7.07	75.31 ± 5.96	74.17 ± 6.34	75.34 ± 7.31	70.04 ± 8.00	**79.12** ± 5.92	70.10 ± 8.46
Tic-tac-toe	78.29 ± 5.07	78.71 ± 3.36	**83.51** ± 3.10	77.87 ± 5.25	73.07 ± 2.70	65.35 ± 3.44	65.35 ± 2.63
Wine	**97.19** ± 5.09	92.68 ± 7.91	94.90 ± 3.30	94.93 ± 3.17	95.49 ± 4.40	96.05 ± 9.24	83.17 ± 4.38
Wisconsin	96.42 ± 1.55	96.13 ± 2.95	95.14 ± 2.62	95.86 ± 2.47	95.57 ± 2.73	**96.71** ± 3.36	95.99 ± 2.72
Yeast	53.37 ± 3.36	**54.18** ± 4.38	52.30 ± 3.94	53.50 ± 3.77	52.23 ± 4.39	**54.18** ± 4.74	53.30 ± 4.02
Zoo	96.39 ± 4.80	94.22 ± 7.94	95.42 ± 6.00	90.72 ± 7.09	**96.50** ± 4.61	89.14 ± 7.88	86.08 ± 4.18
Average	**78.88** ± 5.73	76.93 ± 5.78	76.50 ± 5.87	77.89 ± 5.72	75.24 ± 6.58	73.86 ± 6.17	70.78 ± 6.52
Best result (of 30)	13	4	5	4	2	4	1

Furthermore, median estimators computed by the Contrast Estimation (Table 10) represent the quantitative difference between the different methods considered in the experimental study. Fig. 3 depicts these results graphically. As can be seen in the graph, EIS-RFS achieves a moderate improvement on IFS-SSGA. This improvement is greater when comparing EIS-RFS with IS-SSGA, FS-RST + IS-SSGA and FS-SSGA. The greatest differences are found comparing it with FS-RST and IS-SSGA + FS-RST.

In summary, all these results depict EIS-RFS as an outstanding approach for enhancing the behavior of the K-NN classifier, with respect to the related techniques selected. It offers the most accurate results, while reduction rates are maintained with respect to its related techniques (thus the test phase will take the same computational resources – time and storage requirement – as the rest of methods). Moreover, its computational time in training phase is comparable to the rest of the evolutionary techniques, allowing it to be employed in standard classification problems with ease.

In addiction, it is important to point out that it significantly improves the non-hybrid proposals considered in the experimental framework: FS-RST + IS-SSGA and IS-SSGA + FS-RST. This fact reinforces the validity of the hybrid approach, in contrast to simply using both techniques one after the other.

5.3. Comparison in large domains

Table 11 shows the results measured by accuracy in test data for each method considered in the study with large data sets (including the 1-NN classifier as baseline). For each one, the mean accuracy and the standard deviation are computed. The best case in each data set is highlighted in bold. The last row in each table shows the average considering all data sets.

Tables 12 and 13 reports the average reduction rates achieved and the time elapsed footnote 3, respectively.

Observing the results shown in the tables, we can make the following analysis:

- EIS-RFS again achieves the best accuracy result, obtaining the best overall result in 11 of the 13 data sets. In addition, it still greatly outperforms the 1-NN classifier.
- Concerning reduction in instances' space, all the methods considered achieve a similar rate. Again, the computational time spent by the K-NN classifier in test phase will be very low if any of the reference sets produced is used, including the one preprocessed by EIS-RFS.

Table 7
Average reduction results over instances and features.

Data set	Instances					Features					
	EIS-RFS	IS-SSGA	IFS-SSGA	FS-RST + IS-SSGA	IS-SSGA + FS-RST	EIS-RFS	FS-SSGA	IFS-SSGA	FS-RST	FS-RST + IS-SSGA	IS-SSGA + FS-RST
Australian	0.8872	0.8799	0.8808	0.8808	0.8799	0.1571	0.8071	0.7929	0.0000	0.7929	0.2643
Balance	0.8464	0.8686	0.8085	0.9085	0.8686	0.0000	0.3000	0.0000	0.0000	0.0000	0.5250
Bupa	0.8502	0.8644	0.8644	0.8644	0.8644	0.0000	0.3667	0.4333	0.1274	0.4333	0.0000
Cleveland	0.9014	0.9171	0.9289	0.9289	0.9171	0.0462	0.7385	0.6077	0.3908	0.6077	0.5231
Contraceptive	0.7637	0.7530	0.7530	0.7530	0.7530	0.0667	0.4556	0.5889	0.0360	0.5889	0.1222
Crx	0.8914	0.8816	0.8805	0.8805	0.8816	0.1800	0.5667	0.5533	0.2000	0.5533	0.4067
Ecoli	0.8882	0.9077	0.9130	0.9130	0.9077	0.1286	0.1714	0.1857	0.2286	0.1857	0.2571
Flare-solar	0.8122	0.8391	0.8005	0.8405	0.8391	0.0556	0.5111	0.5778	0.1556	0.5778	0.3000
German	0.8014	0.7914	0.7928	0.7928	0.7914	0.2350	0.5150	0.7450	0.1450	0.7450	0.4900
Glass	0.8718	0.8791	0.8791	0.8791	0.8791	0.0444	0.4444	0.4556	0.0168	0.4556	0.0778
Haberman	0.9306	0.9379	0.9379	0.9379	0.9379	0.0000	0.6667	0.5333	0.0254	0.5333	0.0000
Hayes-roth	0.8544	0.8384	0.8452	0.8452	0.8384	0.2500	0.2500	0.2500	0.1000	0.2500	0.2500
Heart	0.9255	0.9506	0.9230	0.9230	0.9506	0.2308	0.4538	0.5692	0.1846	0.5692	0.6000
Hepatitis	0.9262	0.9226	0.9355	0.9355	0.9226	0.5368	0.6684	0.5421	0.4263	0.5421	0.7211
Housevotes	0.9387	0.9410	0.9653	0.9653	0.9410	0.3500	0.7000	0.7313	0.0188	0.7313	0.8625
Iris	0.9511	0.9481	0.9481	0.9481	0.9481	0.1250	0.4000	0.4500	0.0000	0.4500	0.0000
Led7digit	0.9416	0.9071	0.9491	0.9491	0.9071	0.0000	0.0143	0.0000	0.0143	0.0000	0.8571
Lymphography	0.9257	0.8994	0.9234	0.9234	0.8994	0.4444	0.6500	0.6500	0.2611	0.6500	0.6944
Mammographic	0.8322	0.8229	0.7829	0.8229	0.8229	0.0000	0.5000	0.6200	0.3396	0.6200	0.0000
Monk-2	0.9342	0.8570	0.9406	0.9406	0.8570	0.5000	0.5000	0.5333	0.0000	0.5333	0.5000
Newthyroid	0.9473	0.9571	0.9571	0.9571	0.9571	0.0600	0.3000	0.3800	0.0000	0.3800	0.1000
Pima	0.7911	0.8187	0.8187	0.8187	0.8187	0.0000	0.5750	0.4375	0.0000	0.4375	0.0875
Saheart	0.8668	0.8841	0.8778	0.8778	0.8841	0.0000	0.6333	0.5778	0.0000	0.5778	0.3111
Sonar	0.8899	0.8595	0.8974	0.8974	0.8595	0.2900	0.6633	0.6600	0.7183	0.6600	0.9167
Spectfheart	0.9497	0.9426	0.9409	0.9409	0.9426	0.2727	0.6750	0.6614	0.2750	0.6614	0.8773
Tic-tac-toe	0.8655	0.7917	0.8047	0.8747	0.7917	0.0000	0.2444	0.2889	0.0000	0.2889	0.8889
Wine	0.9451	0.9538	0.9557	0.9557	0.9538	0.3308	0.4538	0.4538	0.5231	0.4538	0.7462
Wisconsin	0.9103	0.9027	0.9048	0.9048	0.9027	0.0444	0.3889	0.3222	0.0000	0.3222	0.3667
Yeast	0.7550	0.7485	0.7485	0.7485	0.7485	0.0375	0.0875	0.1625	0.1256	0.1625	0.2375
Zoo	0.8634	0.8714	0.8468	0.8468	0.8714	0.2125	0.7125	0.3750	0.2750	0.3750	0.7500
Average	0.8819	0.8779	0.8802	0.8885	0.8779	0.1533	0.4804	0.4713	0.1529	0.4713	0.4244

- In the features' space, EIS-RFS obtains a slightly lower result than the rest of the reference techniques. However, with the high reduction rates obtained in the instances' space, the lower reduction rates over features will not cause the K-NN classifier to consume too much time in test phase.
- Concerning time elapsed, we again obtain the same results: EIS-RFS spent a slightly higher time than the rest of the methods, except for FS-RST and 1-NN (whose computational time is not comparable since they only evaluate a single reference set, instead of the 10,000 evaluations performed by the evolutionary methods), and FS-SSGA, whose time requirements are greater than the rest.

This time, the statistical analysis performed is shown in Tables 14 and 15. The former presents the results of the Wilcoxon Signed Ranks test, the latter shows the results of the Contrast Estimation. Again, note that these tests have been carried out considering the average accuracy results obtained.

The Wilcoxon test shows that our approach, EIS-RFS, statistically outperforms all the comparison methods with a level of significance $\alpha = 0.01$; that is, no comparison of EIS-RFS and any comparison method achieves a P-value equal to or higher than 0.01. This result confirms that the good behavior of EIS-RFS is maintained when applied to large data sets.

Median estimators computed by Contrast Estimation (Table 15) represent greater differences than in the former study, although similar conclusions can be drawn. Fig. 4 depicts these results graphically. As can be seen in the graph, EIS-RFS achieves a moderate improvement over the purely evolutionary methods (IS-SSGA, FS-SSGA and IFS-SSGA), which is greater when compared with FS-RST. The greatest differences are found comparing it with and FS-RST + IS-SSGA, IS-SSGA + FS-RST and 1-NN.

The results obtained in this part of the study contrast the quality of EIS-RFS further. Its capabilities are not diminished if it is applied over large data sets, even maintaining the same number of evaluations as in the standard study. Moreover, although its application in these domains is costly, the computational times reported suggest that EIS-RFS can be used in these domains without the necessity of exceptional computer resources, allowing the model to be considered for use in real-world applications.

Table 8
Average time elapsed (training phase), in seconds.

Data set	EIS-RFS	IS-SSGA	FS-SSGA	IFS-SSGA	FS-RST	FS-RST + IS-SSGA	IS-SSGA + FS-RST
Australian	82.54	79.16	161.39	48.14	0.70	81.13	79.90
Balance	54.44	38.71	88.56	38.33	0.03	29.41	37.45
Bupa	20.71	13.70	32.65	11.33	0.04	13.17	13.28
Cleveland	19.29	11.89	33.37	9.04	0.10	10.17	11.93
Contraceptive	316.30	348.66	704.06	306.52	1.32	347.29	339.26
Crx	86.38	79.72	220.59	70.56	0.46	83.90	83.55
Ecoli	20.68	10.92	37.50	11.17	0.05	10.35	11.06
Flare-solar	183.44	160.00	349.09	123.76	0.01	146.04	145.15
German	304.94	252.59	591.00	167.51	2.07	245.40	263.49
Glass	10.30	5.39	15.93	5.18	0.05	5.63	5.56
Haberman	9.41	7.09	13.63	6.06	0.01	7.15	7.07
Hayes-roth	3.86	2.68	5.00	2.52	0.02	2.74	2.77
Heart	14.57	8.03	32.98	7.01	0.06	7.91	8.02
Hepatitis	8.50	3.83	13.08	3.21	0.04	3.43	3.90
Housevotes	39.42	24.98	82.91	17.38	0.02	15.94	25.31
Iris	4.40	2.44	5.22	2.33	0.02	2.55	2.45
Led7digit	40.50	25.05	88.31	28.87	0.00	17.60	25.21
Lymphography	8.14	3.97	11.77	3.23	0.02	3.19	3.99
Mammographic	127.75	116.67	205.34	77.57	0.20	105.70	112.27
Monk-2	27.92	20.72	46.18	14.22	0.02	14.06	20.24
Newthyroid	8.03	3.68	11.76	3.63	0.01	3.81	3.66
Pima	96.68	85.38	175.95	72.09	0.29	77.98	78.76
Saheart	33.45	25.98	62.64	22.45	0.15	24.01	25.78
Sonar	136.71	17.12	66.79	13.86	0.40	9.91	17.22
Spectfheart	40.33	15.14	80.53	15.25	0.27	12.10	15.23
Tic-tac-toe	176.75	150.31	348.57	132.37	0.06	75.92	138.99
Wine	7.67	3.62	14.80	3.34	0.03	3.40	3.64
Wisconsin	73.61	55.66	159.05	50.62	0.10	49.54	53.67
Yeast	420.58	354.55	825.97	364.96	1.47	351.06	342.82
Zoo	5.31	2.52	5.12	2.46	0.01	2.33	2.46
Average	79.42	64.34	149.66	54.50	0.27	58.76	62.80

Table 9
Wilcoxon signed-ranks test results.

Comparison	R^+	R^-	P-value
EIS-RFS vs IS-SSGA	381	84	0.00158
EIS-RFS vs FS-SSGA	342	93	0.00599
EIS-RFS vs IFS-SSGA	371.5	93.5	0.00335
EIS-RFS vs FS-RST	426	39	0.00001
EIS-RFS vs FS-RST + IS-SSGA	389	39	0.00007
EIS-RFS vsIS-SSGA + FS-RST	456	39	0.00000

Table 10
Contrast estimation results.

EIS-RFS vs	Median estimation
IS-SSGA	1.031
FS-SSGA	1.946
IFS-SSGA	0.435
FS-RST	2.941
FS-RST + IS-SSGA	1.706
IS-SSGA + FS-RST	3.966

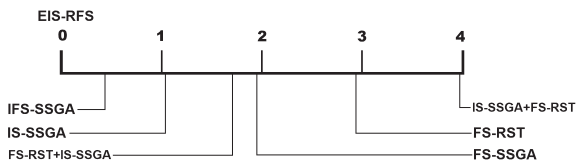

Fig. 3. Graphic depicting the Contrast Estimation of medians between EIS-RFS and the comparison methods.

Table 11
Accuracy results in test phase.

Data set	EIS-RFS	IS-SSGA	FS-SSGA	IFS-SSGA	FS-RST	FS-RST + IS-SSGA	IS-SSGA + FS-RST	1-NN
Abalone	**26.31** ± 2.01	21.78 ± 1.96	20.82 ± 1.98	20.44 ± 1.81	19.84 ± 1.89	19.66 ± 2.01	12.87 ± 2.17	10.38 ± 1.74
Banana	**89.34** ± 2.70	88.42 ± 0.95	87.49 ± 1.05	88.06 ± 1.38	87.49 ± 1.05	81.70 ± 2.78	78.48 ± 2.88	74.76 ± 2.08
Chess	**91.30** ± 5.38	86.83 ± 2.36	64.23 ± 0.76	67.03 ± 1.06	75.47 ± 2.07	73.23 ± 5.79	74.86 ± 5.65	68.99 ± 4.84
Marketing	**30.12** ± 1.85	27.10 ± 1.15	27.02 ± 1.14	29.09 ± 1.70	27.58 ± 1.06	24.90 ± 2.00	18.35 ± 1.87	17.20 ± 1.52
Mushroom	**100.00** ± 0.00	99.96 ± 0.08	**100.00** ± 0.00	99.94 ± 0.13	98.52 ± 0.35	98.26 ± 0.55	99.00 ± 0.34	**100.00** ± 0.00
Page-blocks	95.25 ± 5.51	94.97 ± 0.83	**96.44** ± 0.73	95.63 ± 0.44	95.07 ± 0.76	95.10 ± 5.72	85.48 ± 6.01	76.73 ± 5.84
Ring	**86.01** ± 1.86	74.93 ± 1.47	83.01 ± 1.38	82.01 ± 1.08	79.18 ± 1.35	71.39 ± 1.98	61.39 ± 1.94	50.25 ± 1.64
Satimage	**89.60** ± 1.87	87.73 ± 1.65	89.44 ± 1.96	88.65 ± 1.68	85.21 ± 1.99	84.36 ± 1.95	83.30 ± 1.97	88.38 ± 1.63
Segment	94.37 ± 0.94	94.59 ± 1.46	**96.84** ± 1.02	95.93 ± 1.21	94.29 ± 1.61	94.92 ± 1.03	93.49 ± 0.99	96.06 ± 0.82
Spambase	**89.35** ± 3.05	82.60 ± 0.72	83.47 ± 1.40	87.54 ± 2.07	81.74 ± 1.74	76.93 ± 3.23	79.43 ± 3.20	77.89 ± 2.49
Splice	**83.07** ± 2.32	73.57 ± 1.36	76.93 ± 2.61	72.95 ± 2.49	85.89 ± 1.85	72.46 ± 2.38	58.26 ± 2.40	60.55 ± 1.85
Titanic	**79.38** ± 11.23	78.78 ± 2.33	76.10 ± 5.60	78.74 ± 2.50	59.16 ± 8.08	51.35 ± 11.61	61.44 ± 11.94	13.98 ± 12.33
Twonorm	**96.54** ± 1.34	95.54 ± 1.64	94.86 ± 1.76	94.59 ± 1.83	77.86 ± 1.72	80.73 ± 1.37	81.10 ± 1.47	89.35 ± 1.46
Average	**80.82** ± 3.08	77.45 ± 1.38	76.67 ± 1.65	76.97 ± 1.49	74.41 ± 1.96	71.15 ± 3.26	68.27 ± 3.29	63.42 ± 2.94
Best result (of 13)	11	0	3	0	0	0	0	1

Table 12
Average reduction results over instances and features.

Data set	Instances					Features					
	EIS-RFS	IS-SSGA	IFS-SSGA	FS-RST + IS-SSGA	IS-SSGA + FS-RST	EIS-RFS	FS-SSGA	IFS-SSGA	FS-RST	FS-RST + IS-SSGA	IS-SSGAFS-RST
Abalone	0.7401	0.7426	0.7391	0.6913	0.7426	0.3575	0.5875	0.4625	0.5000	0.5000	0.5000
Banana	0.7509	0.7560	0.7483	0.7623	0.7560	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Chess	0.7466	0.7578	0.7686	0.7184	0.7578	0.4722	0.4333	0.5306	0.9444	0.9444	0.5820
Marketing	0.7420	0.7357	0.7341	0.7326	0.7357	0.3585	0.7000	0.6385	0.3846	0.3846	0.3639
Mushroom	0.7620	0.7619	0.7609	0.7244	0.7619	0.8383	0.8091	0.5955	0.9545	0.9545	0.9121
Page-blocks	0.7643	0.7661	0.7610	0.7631	0.7661	0.7922	0.6000	0.4300	0.7000	0.7000	0.7000
Ring	0.7308	0.7258	0.7344	0.7345	0.7258	0.4537	0.5050	0.5450	0.7000	0.7000	0.6350
Satimage	0.7498	0.7528	0.7461	0.7020	0.7528	0.5700	0.5556	0.4611	0.8611	0.8611	0.6140
Segment	0.8062	0.8053	0.7993	0.7731	0.8053	0.0000	0.6895	0.6737	0.0000	0.0000	0.0000
Spambase	0.7549	0.7538	0.7389	0.7349	0.7538	0.5584	0.5298	0.5930	0.8596	0.8596	0.6930
Splice	0.7572	0.7431	0.7428	0.6913	0.7431	0.6833	0.8317	0.8317	0.8667	0.8667	0.8333
Titanic	0.8299	0.8283	0.8202	0.8156	0.8283	0.0000	0.5333	0.0333	0.0000	0.0000	0.0000
Twonorm	0.7521	0.7557	0.7285	0.7194	0.7557	0.0000	0.0000	0.0100	0.7500	0.7500	0.7500
Average	0.7605	0.7604	0.7556	0.7356	0.7604	0.3911	0.5211	0.4465	0.5785	0.5785	0.5064

Table 13
Average time elapsed (training phase), in seconds.

Data set	EIS-RFS	IS-SSGA	FS-SSGA	IFS-SSGA	FS-RST	FS-RST + IS-SSGA	IS-SSGA + FS-RST
Abalone	3739	3546	7992	3265	27	3391	3436
Banana	2496	3145	6992	3343	2	2948	3099
Chess	14345	5614	20910	4794	1	5249	5456
Marketing	21471	27793	47621	19468	98	25989	27774
Mushroom	33249	33121	62833	23482	4	31833	33136
Page-blocks	5853	7942	16681	6668	14	7419	7741
Ring	26797	24125	59894	20742	325	23185	24401
Satimage	62529	26160	82022	27992	426	25574	27607
Segment	1384	1641	4891	1206	4	1567	1618
Spambase	41759	19965	63763	19872	560	18870	19472
Splice	62673	9431	26504	5898	395	8951	9376
Titanic	413	605	1402	516	0	573	603
Twonorm	24515	22273	70736	25004	292	20941	21865
Average	23171	14259	36326	12481	165	13576	14276

Table 14
Wilcoxon signed-ranks test results (large data sets).

Comparison	R^+	R^-	P-value
EIS-RFS vs IS-SSGA	89	2	0.00073
EIS-RFS vs FS-SSGA	71	7	0.00928
EIS-RFS vs IFS-SSGA	82	9	0.00806
EIS-RFS vs FS-RST	85	6	0.00342
EIS-RFS vs FS-RST + IS-SSGA	89	2	0.00073
EIS-RFS vs IS-SSGA + FS-RST	91	0	0.00024
EIS-RFS vs 1-NN	76	2	0.00146

Table 15
Contrast estimation results.

EIS-RFS vs	Median estimation
IS-SSGA	2.653
FS-SSGA	2.849
IFS-SSGA	2.598
FS-RST	4.855
FS-RST + IS-SSGA	7.260
IS-SSGA + FS-RST	10.784
1-NN	13.540

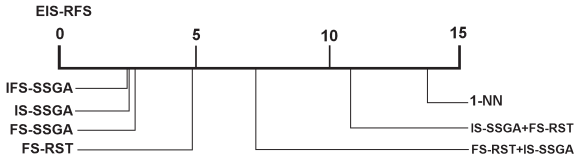

Fig. 4. Graphic depicting the Contrast Estimation of medians between EIS-RFS and the comparison methods in large data sets.

6. Conclusions

In this paper, we have presented EIS-RFS, a novel approach which introduces the cooperation between two well-known techniques for data reduction: A steady-state GA for IS, and a fuzzy RST based method for FS. The judicious selection of features performed by the RST based method has been shown to be beneficial for the search procedure performed by the GA, thus allowing our approach to achieve highly reduced training sets which greatly enhances the behavior of the K-NN classifier.

The experimental study performed has highlighted these improvements, especially in the case of 1-NN. It has also shown that EIS-RFS outperforms several preprocessing methods related to IS and FS, including hybrid models, considering classification accuracy. Moreover, these results are maintained when it is applied to higher size data sets, thus confirming

the capabilities of our approach as a suitable preprocessing method for most of the standard problems present in supervised classification.

As regards future work, we point out the possibility of using EIS-RFS to enhance other kinds of machine learning algorithms. Given the nature of our method (in essence, a wrapper-based model) and the generality of the fuzzy rough feature selection method and the fitness function defined, it is possible to apply it to improve the results of the majority of machine learning methods. The only requirement would be the existence of a way in which the current performance of the method could be evaluated such as, for example, the accuracy measure of the K-NN classifier. A suitable starting point for this line would be the research shown in [8,9,37], where IS (namely TSS) is used to enhance other models such as decision trees, subgroup discovery methods and neural networks, respectively.

Acknowledgements

This work was supported by Project TIN2008-06681-C06-01. J. Derrac holds an FPU scholarship from Spanish Ministry of Education. Chris Cornelis would like to thank the Research Foundation – Flanders for funding his research.

Appendix A. A nonparametric method for analyzing medians of classifiers: the contrast estimation

The Contrast Estimation based on medians [22] can be used to estimate the difference between two classifiers' performance. It assumes that the expected differences between performances of algorithms are the same across data sets. Therefore, the performance of methods is reflected by the magnitudes of the differences between them in each domain.

The interest of this test lies in estimating the contrast between medians of samples of results considering all pairwise comparisons. The test obtains a quantitative difference computed through medians between two algorithms over multiple data sets, proceeding as follows:

1. For every pair of k algorithms in the experiment, compute the difference between the performances x of the two algorithms in each of the n data sets. That is, compute the differences
$$D_{i(u,v)} = x_{iu} - x_{iv} \tag{A.1}$$
where $i = 1,\ldots,n$; $u = 1,\ldots,k$; $v = 1,\ldots,k$. (consider only performance pairs where $u < v$).
2. Find the median of each set of differences (Z_{uv}, which can be regarded as the *unadjusted estimator* of the medians of the methods u and v, $M_u - M_v$). Since $Z_{uv} = Z_{vu}$, it is only required to compute Z_{uv} in those cases where $u < v$. Also note that $Z_{uu} = 0$.
3. Compute the mean of each set of unadjusted medians having the same first subscript, m_u:
$$m_u = \frac{\sum_{j=1}^{k} Z_{uj}}{k}, u = 1,\ldots,k \tag{A.2}$$
4. The estimator of $M_u - M_v$ is $m_u - m_v$, where u and v range from 1 through k. For example, the difference between M_1 and M_2 is estimated by $m_1 - m_2$

These estimators can be understood as an advanced global performance measure. Although this test cannot provide a probability of error associated with the rejection of the null hypothesis of equality, it is especially useful to estimate how far a method outperforms another one.

An implementation of the Contrast Estimation procedure can be found in the CONTROLTEST package, which can be obtained at the SCI2S thematic public website on *Statistical Inference in Computational Intelligence and Data Mining* (http://sci2s.ugr.es/sicidm/).

References

[1] D.W. Aha (Ed.), Lazy Learning, Springer, 2009.
[2] H. Ahn, K. Kim, Bankruptcy prediction modeling with hybrid case-based reasoning and genetic algorithms approach, Applied Soft Computing 9 (2009) 599–607.
[3] 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, Journal of Multiple-Valued Logic and Soft Computing 17 (2011).
[4] J. Alcalá-Fdez, L. Sánchez, S. García, M.J. del Jesus, S. Ventura, J.M. Garrell, J. Otero, C. Romero, J. Bacardit, V.M. Rivas, J.C. Fernández, F. Herrera, KEEL: a software tool to assess evolutionary algorithms for data mining problems, Soft Computing 13 (2008) 307–318.
[5] E. Alpaydin, Introduction to Machine Learning, second ed., The MIT Press, 2010.
[6] G. Bell, T. Hey, A. Szalay, Beyond the data deluge, Science 323 (2009) 1297–1298.
[7] J.R. Cano, F. Herrera, M. Lozano, Using evolutionary algorithms as instance selection for data reduction in KDD: An experimental study, IEEE Transactions on Evolutionary Computation 7 (2003) 561–575.
[8] J.R. Cano, F. Herrera, M. Lozano, Evolutionary stratified training set selection for extracting classification rules with trade-off precision-interpretability, Data and Knowledge Engineering 60 (2007) 90–100.
[9] J.R. Cano, F. Herrera, M. Lozano, S. García, Making cn2-sd subgroup discovery algorithm scalable to large size data sets using instance selection, Expert Systems with Applications 35 (2008) 1949–1965.

[10] D. Chen, C. Wang, Q. Hu, A new approach to attribute reduction of consistent and inconsistent covering decision systems with covering rough sets, Information Sciences 177 (2007) 3500v–3518.
[11] Y. Chen, E.K. Garcia, M.R. Gupta, A. Rahimi, L. Cazzanti, Similarity-based classification: Concepts and algorithms, Journal of Machine Learning Research 10 (2009) 747–776.
[12] C. Cornelis, R. Jensen, G. Hurtado, D. Slezak, Attribute selection with fuzzy decision reducts, Information Sciences 180 (2010) 209–v224.
[13] T.M. Cover, P.E. Hart, Nearest neighbor pattern classification, IEEE Transactions on Information Theory 13 (1967) 21–27.
[14] J. Derrac, S. García, F. Herrera, IFS-CoCo: Instance and feature selection based on cooperative coevolution with nearest neighbor rule, Pattern Recognition 43 (2010) 2082–2105.
[15] J. Derrac, S. García, F. Herrera, A survey on evolutionary instance selection and generation, International Journal of Applied Metaheuristic Computing 1 (2010) 60–92.
[16] J. Derrac, S. García, D. Molina, F. Herrera, A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms, Swarm and Evolutionary Computation 1 (2011) 3–18.
[17] A.E. Eiben, J.E. Smith, Introduction to evolutionary computing, Natural Computing, Springer-Verlag, 2003.
[18] A. Frank, A. Asuncion, UCI machine learning repository, 2010.
[19] A.A. Freitas, Data Mining and Knowledge Discovery with Evolutionary Algorithms, Springer-Verlag, 2002.
[20] S. García, J.R. Cano, F. Herrera, A memetic algorithm for evolutionary prototype selection: A scaling up approach, Pattern Recognition 41 (2008) 2693–2709.
[21] S. García, J. Derrac, J.R. Cano, F. Herrera, Prototype selection for nearest neighbor classification: Taxonomy and empirical study, IEEE Transactions on Pattern Analysis and Machine Intelligence, in press, doi:10.1109/TPAMI.2011.142.
[22] 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, Information Sciences 180 (2010) 2044–2064.
[23] S. García, F. Herrera, An extension on Statistical Comparisons of Classifiers over Multiple Data Sets for all pairwise comparisons, Journal of Machine Learning Research 9 (2008) 2677–2694.
[24] N. García-Pedrajas, J.A.R. del Castillo, D. Ortiz-Boyer, A cooperative coevolutionary algorithm for instance selection for instance-based learning, Machine Learning 78 (2010) 381–420.
[25] A. Ghosh, L.C. Jain (Eds.), Evolutionary Computation in Data Mining, Springer-Verlag, 2005.
[26] R. Gil-Pita, X. Yao, Evolving edited k-nearest neighbor classifiers, International Journal of Neural Systems 18 (2008) 1–9.
[27] I. Guyon, A. Elisseeff, An introduction to variable and feature selection, Journal of Machine Learning Research 3 (2003) 1157–1182.
[28] I. Guyon, S. Gunn, M. Nikravesh, L.A. Zadeh (Eds.), Feature Extraction: Foundations and Applications, Springer, 2006.
[29] S.Y. Ho, C.C. Liu, S. Liu, Design of an optimal nearest neighbor classifier using an intelligent genetic algorithm, Pattern Recognition Letters 23 (2002) 1495–1503.
[30] Q. Hu, X. Xie, D. Yu, Hybrid attribute reduction based on a novel fuzzy-rough model and information granulation, Pattern Recognition 40 (2007) 3509–3521.
[31] I. Inza, P. Larrañaga, B. Sierra, Feature subset selection by bayesian networks: a comparison with genetic and sequential algorithms, International Journal of Approximate Reasoning 27 (2001) 143–164.
[32] H. Ishibuchi, T. Nakashima, Evolution of reference sets in nearest neighbor classification, in: Second Asia-Pacific Conference on Simulated Evolution and Learning on Simulated Evolution and Learning (SEAL'98), vol. 1585, Lecture Notes in Computer Science, 1998, pp. 82–89.
[33] M.Z. Jahromi, E. Parvinnia, R. John, A method of learning weighted similarity function to improve the performance of nearest neighbor, Information Sciences 179 (2009) 2964–2973.
[34] R. Jensen, C. Cornelis, Fuzzy-rough instance selection, in: Proceedings of the WCCI 2010 IEEE World Congress on Computational Intelligence, IEEE Congress on Fuzzy Logic, Barcelona Spain, 2010, pp. 1776–1782.
[35] R. Jensen, Q. Shen, Fuzzy-rough sets assisted attribute selection, IEEE Transactions on Fuzzy Systems 15 (2007) 73–89.
[36] R. Jensen, Q. Shen, New approaches to fuzzy-rough feature selection, IEEE Transactions on Fuzzy Systems 17 (2009) 824–838.
[37] K. Kim, Artificial neural networks with evolutionary instance selection for financial forecasting, Expert Systems with Applications 30 (2006) 519–526.
[38] L.I. Kuncheva, Editing for the *k*-nearest neighbors rule by a genetic algorithm, Pattern Recognition Letters 16 (1995) 809–814.
[39] H. Liu, F. Hussain, C.L. Tan, M. Dash, Discretization: An enabling technique, Data Mining and Knowledge Discovery 6 (2002) 393–423.
[40] H. Liu, H. Motoda (Eds.), Feature selection for knowledge discovery and data mining, The Springer International Series in Engineering and Computer Science, Springer, 1998.
[41] H. Liu, H. Motoda (Eds.), Instance selection and construction for data mining, The Springer International Series in Engineering and Computer Science, Springer, 2001.
[42] H. Liu, H. Motoda, On issues of instance selection, Data Mining and Knowledge Discovery 6 (2002) 115–130.
[43] H. Liu, H. Motoda (Eds.), Computational methods of feature selection, Chapman & Hall/Crc Data Mining and Knowledge Discovery Series, Chapman & Hall/Crc, 2007.
[44] H. Liu, L. Yu, Toward integrating feature selection algorithms for classification and clustering, IEEE Transactions on Knowledge and Data Engineering 17 (2005) 1–12.
[45] E. Mjolsness, D. DeCoste, Machine learning for science: State of the art and future prospects, Science 293 (2001) 2051–2055.
[46] H.S. Nguyen, A. Skowron, J. Stepaniuk, Granular computing: A rough set approach, Computational Intelligence 17 (2001) 514–544.
[47] I.S. Oh, J.S. Lee, B.R. Moon, Hybrid genetic algorithms for feature selection, IEEE Transactions on Pattern Analysis and Machine Intelligence 26 (2004) 1424–1437.
[48] A.N. Papadopoulos, Y. Manolopoulos, Nearest Neighbor Search: A Database Perspective, Springer Verlag Telos, 2004.
[49] G.L. Pappa, A.A. Freitas, Automating the design of data mining algorithms: an evolutionary computation approach, Natural Computing, Springer, 2009.
[50] Z. Pawlak, Rough sets, International Journal of Computer and Information Sciences 11 (1982) 341v–356.
[51] Z. Pawlak, Rough Sets: Theoretical Aspects of Reasoning About Data, Kluwer Academic Publishing, 1991.
[52] Z. Pawlak, A. Skowron, Rough sets: some extensions, Information Sciences 177 (2007) 28–40.
[53] Z. Pawlak, A. Skowron, Rudiments of rough sets, Information Sciences 177 (2007) 3–27.
[54] D. Pyle, Data preparation for data mining, The Morgan Kaufmann Series in Data Management Systems, Morgan Kaufman, 1999.
[55] G.J.R. Kohavi, Wrappers for feature selection, Artificial Intelligence 97 (1997) 273–324.
[56] A. Radzikowska, E. Kerre, A comparative study of fuzzy rough sets, Fuzzy Sets and Systems 126 (2002) 137–156.
[57] Y. Saeys, I. Inza, P. Larrañaga, A review of feature selection techniques in bioinformatics, Bioinformatics 19 (2007) 2507–2517.
[58] G. Shakhnarovich, T. Darrell, P. Indyk (Eds.), Nearest-Neighbor Methods in Learning and Vision: Theory and Practice, The MIT Press, 2006.
[59] D.J. Sheskin, Handbook of Parametric and Nonparametric Statistical Procedures, fifth ed., Chapman & Hall/CRC, 2011.
[60] A. Skowron, J. Stepaniuk, Tolerance approximation spaces, Fundamenta Informaticae 27 (1996).
[61] A. Skowron, J. Stepaniuk, R.W. Swiniarski, Approximation spaces in rough-granular computing, Fundamenta Informaticae 100 (2010).
[62] R. Slowinski, D. Vanderpooten, A generalized definition of rough approximations based on similarity, IEEE Transactions on Knowledge and Data Engineering 12 (2000).
[63] C. Stanfill, D. Waltz, Toward memory-based reasoning, Communications of the ACM 29 (1986) 1213–1228.
[64] M. Stone, Cross-validatory choice and assessment of statistical predictions (with discussion), Journal of the Royal Statistical Society B 36 (1974) 111v–147.

[65] I. Triguero, J. Derrac, S. García, F. Herrera, A taxonomy and experimental study on prototype generation for nearest neighbor classification, IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews, in press, doi: 10.1109/TSMCC.2010.2103939.
[66] I. Triguero, S. García, F. Herrera, IPADE: Iterative prototype adjustment for nearest neighbor classification, IEEE Transactions on Neural Networks 21 (2010) 1984–1990.
[67] E. Tsang, D. Chen, D. Yeung, X. Wang, J.T. Lee, Attributes reduction using fuzzy rough sets, IEEE Transactions on Fuzzy Systems 16 (2008) 1130–1141.
[68] K. Weinberger, L. Saul, Distance metric learning for large margin nearest neighbor classification, Journal of Machine Learning Research 10 (2009) 207–244.
[69] F. Wilcoxon, Individual comparisons by ranking methods, Biometrics Bulletin 1 (1945) 80–83.
[70] D. Wilson, T. Martinez, Improved heterogeneous distance functions, Journal of Artificial Intelligence Research 6 (1997) 1–34.
[71] D.R. Wilson, T.R. Martinez, Reduction techniques for instance-based learning algorithms, Machine Learning 38 (2000) 257–286.
[72] I.H. Witten, E. Frank, Data Mining: Practical Machine Learning Tools and Techniques, second ed., Morgan Kaufmann Series in Data Management Systems, Morgan Kaufmann, 2005.
[73] X. Wu, V. Kumar (Eds.), The top ten algorithms in data mining, Data Mining and Knowledge Discovery, Chapman & Hall/CRC, 2009.
[74] X. Yang, J. Yang, C. Wu, D. Yu, Dominance-based rough set approach and knowledge reductions in incomplete ordered information system, Information Sciences 178 (2008) 1219–1234.
[75] L.A. Zadeh, Fuzzy sets, Information and Control 8 (1965) 338–353.