Soft Comput (2013) 17:223–238
DOI 10.1007/s00500-012-0888-3

FOCUS

On the use of evolutionary feature selection for improving fuzzy rough set based prototype selection

J. Derrac · N. Verbiest · S. García · C. Cornelis · F. Herrera

Published online: 20 July 2012

Abstract The *k*-nearest neighbors classifier is a widely used classification method that has proven to be very effective in supervised learning tasks. In this paper, a fuzzy rough set method for prototype selection, focused on optimizing the behavior of this classifier, is presented. The hybridization with an evolutionary feature selection method is considered to further improve its performance, obtaining a competent data reduction algorithm for the 1-nearest neighbors classifier. This hybridization is performed in the training phase, by using the solution of each preprocessing technique as the starting condition of the other one, within a cycle. The results of the experimental study, which have been contrasted through nonparametric statistical tests, show that the new hybrid approach obtains very promising results with respect to classification accuracy and reduction of the size of the training set.

Keywords Prototype selection · Feature selection · Data reduction · Fuzzy rough sets · Evolutionary algorithms · Nearest neighbor

J. Derrac (✉) · C. Cornelis · F. Herrera
Department of Computer Science and Artificial Intelligence, CITIC-UGR (Research Center on Information and Communications Technology), University of Granada, 18071 Granada, Spain
e-mail: jderrac@decsai.ugr.es

N. Verbiest · C. Cornelis
Department of Applied Mathematics and Computer Science, Ghent University, Ghent, Belgium

S. García
Department of Computer Science, University of Jaén, 23071 Jaén, Spain

1 Introduction

Supervised classification is one of the most useful techniques in machine learning (Mjolsness and DeCoste 2001; Alpaydin 2010; Witten et al. 2011). Categorizing new objects using data stored in a given training set has become a critical task in many real-world applications of data mining and pattern recognition.

The *k*-nearest neighbors classifier (*k*-NN) (Cover and Hart 1967; Shakhnarovich et al. 2006) is one of the most relevant algorithms in data mining (Wu and Kumar 2009). 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 *k*-NN relies mainly on the quality of the training data (Aha et al. 1991). Furthermore, 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 instances in the data set and the number of features which will be used in the computation of the similarity (distance) function (Chen et al. 2009; Weinberger and Saul 2009).

However, the overwhelming amount of data available nowadays in any field of research (Bell et al. 2009) poses new problems when using the *k*-NN classifier. Gathering, understanding and processing such data often requires the use of advanced tools for managing the represented knowledge in a suitable way. In this sense, many approaches have been proposed to improve the performance of *k*-NN (Triguero et al. 2010; Destercke 2012). Some of the most effective ones work directly over the training data, instead of modifying the computation of the *k*-NN rule. They preprocess the initially available data, aiming to improve the algorithms in terms of efficiency and efficacy.

Data reduction (Pyle 1999) is a data preprocessing task whose main objective is to reduce the original training set.

By removing noisy and irrelevant data—harmful for the majority of machine learning methods—data reduction can help to avoid excessive storage and time requirements, easing and enabling machine learning techniques to deal with large data sets. The best known data reduction processes are feature selection (FS) (Liu and Motoda 2007), feature generation/extraction (Guyon et al. 2006), attribute discretization (García et al. 2012b), prototype generation (Triguero et al. 2012) and prototype selection (PS) (García et al. 2012a).

One of the most successful families of data reduction methods has been originated by evolutionary computation (Eiben and Smith 2003; Ghosh and Jain 2005). Evolutionary algorithms, search algorithms inspired by natural populations to evolve solutions, have been applied to different data reduction problems, modeling them as combinatorial problems (Freitas 2002; Cano et al. 2003; Pappa and Freitas 2009). A remarkable number of evolutionary data reduction techniques have been focused on optimizing the *k*-NN rule.

Fuzzy sets (Zadeh 1965) and rough sets (Pawlak 1982; Pawlak and Skowron 2007b) address two important, complementary characteristics of imperfect data and knowledge: the former model vague information by expressing that objects belong to a set or relation to a given degree, while the latter provide approximations of concepts in the presence of incomplete information (Kusunoki and Inuiguchi 2010). A hybrid fuzzy rough set model was first proposed in (Dubois and Prade 1990), later extended and/or modified by many authors, being applied successfully in various domains (Radzikowska and Kerre 2002; De Cock et al. 2007; Tsang et al. 2008; He and Wu 2011; Zhai 2011). Possibly, the most notable capability of these extensions is that they enable practitioners to apply rough sets analysis directly over data sets representing continuous data, in contrast with pure rough sets methods, which cannot be applied over continuous data sets without discretizing them at a previous step.

In this paper, a hybrid model for data reduction combining fuzzy rough sets and evolutionary algorithms is proposed. A PS algorithm is developed by estimating the quality of every training instance through a fuzzy rough set based quality measure. The solutions obtained by this method are used to adjust the search of an evolutionary FS algorithm. Both the PS and the FS algorithms are applied in succession, using the 1-NN classifier as a wrapper for evaluating the solutions. At the end, the subsets of features and prototypes selected are gathered to generate a final data set, which is used in the classification phase as reference.

We have tested our approach (which we have termed EFS-RPS, evolutionary feature selection for fuzzy rough set based prototype selection) in a wide selection of supervised classification domains, considering 38 different problems. The results have been contrasted by using several non-parametric statistical tests (Sheskin 2011), reinforcing the conclusions obtained in the experiments.

The rest of the paper is organized as follows: Sect. 2 shows an introduction to the main topics of the study. Section 3 describes EFS-RPS and its main characteristics. Section 4 details the experimental study performed to test the performance of the new technique. Finally, Sect. 5 sums up our main conclusions.

2 Data reduction preliminaries

This section gives some preliminaries on data reduction techniques, fuzzy rough sets and evolutionary algorithms:

- Section 2.1 describes feature selection and its application for enhancing the *k*-NN rule.
- Section 2.2 surveys prototype selection and some notable characteristics of the field.
- Section 2.3 recalls some definitions of fuzzy rough sets and various works related to data reduction.
- Section 2.4 reviews the use of evolutionary algorithms for the reduction of training sets in *k*-NN based classifiers.

Throughout the section, the following definitions will be used:

- The data set X consists of N instances which are defined by a set A of M attributes (features) in an M-dimensional space and a class (decision) attribute c. Attribute values in M should be normalized in the interval [0, 1].
- Each instance X_p is defined by $X_p = (X_{p1}, X_{p2}, ..., X_{pM}, X_{pc})$, where X_{pi} is the value of the i-th feature of the p-th instance. The class attribute of the instance is determined by X_{pc}, which means that X_p belongs to the class c.
- The data set is split into two different subsets: A training set TR and a test set TS. After the application of a data reduction algorithm (a PS or a FS one) over TR, a reference set RS $\subseteq$ TR is obtained.

2.1 Feature selection

One of the main data reduction techniques is FS. Its goal is to select the most appropriate subset of features from the initial data set. It aims to eliminate irrelevant and redundant features to obtain a simple and accurate classification system (Liu and Motoda 2007).

FS can be defined as follows: Given a data set composed by TR and TS, a FS algorithm searches for a subset of features $B \subseteq A$. The RS set is built from TR, considering only the features selected in B. Instances from TS are then classified by a data mining algorithm using RS as reference.

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 (Kohavi and John 1997).
- *Filtering methods*, where the selection criterion is independent of the learning algorithm. In these methods, the selection is guided by data related measures (for example, separability measures) (Guyon and Elisseeff 2003).
- *Embedded methods*, where the search for an optimal subset of features is performed within the classifier construction (Saeys et al. 2007).

The most popular algorithms for FS are the classical sequential ones. Forward sequential and backward sequential selection (Liu and Motoda 1998) are the best-known ones. They begin with a feature subset and sequentially add or remove features if they improve the quality of the selection until the algorithm finishes. Other remarkable FS methods are FOCUS (Almuallim and Dietterich 1991) and the RELIEF family (Kira and Rendell 1992).

Despite the popularity of these classical methods, many other approaches based on heuristic search can be found in the literature (Stracuzzi and Utgoff 2004; Shie and Chen 2008). Complete surveys, analyzing both classical and advanced approaches to FS, can be found in the literature (Guyon and Elisseeff 2003; Liu and Yu 2005; Saeys et al. 2007).

2.2 Prototype selection

PS methods are data reduction methods whose objective is to isolate the smallest set of instances which enable the k-NN classifier to predict the class of a query instance with the same quality as the initial data set (Liu and Motoda 2001).

PS can be defined as follows: Given a data set composed by TR and TS, a PS algorithm obtains prototypes as a subset of instances RS $\subseteq$ TR. Instances from TS are then classified by the k-NN classifier using RS as reference.

Depending on the strategy followed, PS methods can be categorized into three classes: *preservation methods*, which aim to obtain a consistent subset from the training data, ignoring the presence of noise; *noise removal methods*, which aim to remove noise both in the boundary points (instances near to the decision boundaries) and in the inner points (instances far from the decision boundaries), and *hybrid methods*, which perform both objectives simultaneously (García et al. 2012a).

PS methods are sometimes dependent on the k value set on the definition of the k-NN classifier. In (Wilson and Martinez 2000), it is stated that setting $k > 1$ decreases the sensitivity of the algorithm to noise and tends to smooth the decision boundaries. In some PS 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 in the reduction process, and then recompute the best value of k in the classification phase. In this work, we have employed the value $k = 1$, to give the classifier the greatest possible sensitivity to noise during the reduction process. In this manner, an evolutionary PS algorithm can detect better the noisy instances and the redundant ones present in the training set.

Despite the variety of PS methods developed in the last decades (with some remarkable proposals such as CNN (Hart 1968), ENN (Wilson 1972) or the IB (Aha et al. 1991) and DROP (Wilson and Martinez 2000) families), improvements in storage reduction, noise tolerance, generalization accuracy and time requirements are reported still nowadays, with the development of new PS methods (García et al. 2008; Ferrandiz and Boullé 2010; Franco et al. 2010; Quirino et al. 2010). They have become a proof of the topical nature of this field, which continues to attract the interest of many research communities in the search for new ways to further improve the performance of the k-NN classifier. More information about the PS field can be found at the SCI2S thematic public website on *Prototype Reduction in Nearest Neighbor Classification*.[1]

2.3 Fuzzy rough sets for data reduction

Rough set theory (Pawlak 1991; Pawlak and Skowron 2007a) provides a methodology for data analysis based on the approximation of concepts in a decision system $(X, A \cup \{c\})$, in which X is a set of instances, A is a set of conditional attributes and c is the decision or class attribute.

The theory revolves around the notion of (in)discernibility: the ability to distinguish between instances, based on their attribute values. When fuzzy rough sets are used, indiscernibility is typically modeled by means of a fuzzy tolerance relation R in X. In this paper, R is defined as, for X_x and X_y in $X(x, y \in \{1, \ldots, n\})$,

$$R(X_x, X_y) = \mathscr{T}(\underbrace{R_a(X_x, X_y)}_{a \in A}) \tag{1}$$

where $\mathscr{T}$ is a t-norm, which is an associative, commutative mapping $\mathscr{T} : [0,1]^2 \rightarrow [0,1]$, increasing in both arguments and such that $\forall s \in [0,1] : \mathscr{T}(s,1) = s$. In this paper we will use the Łukasiewicz t-norm, defined as follows: $\mathscr{T}(s,t) = \max(0, s + t - 1)$ for $s, t \in [0,1]$. Note that, as a t-norm is associative and commutative, it can be extended unambiguously for M arguments as in Eq. (1).

The indiscernibility for one attribute $R_a(X_x, X_y)$ is given by

[1] http://sci2s.ugr.es/pr/.

$$R_a(X_x, X_y) = 1 - (X_{xa} - X_{ya})^2 \quad (2)$$

if a is a quantitative (real) attribute, and

$$R_a(X_x, X_y) = \begin{cases} 1 & \text{if } X_{xa} = X_{ya} \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

when a is nominal (discrete).

Given R, the fuzzy-rough positive region of X is defined as the fuzzy set POS_A in X

$$\text{POS}_\text{A}(X_x) = \min_{X_y \in X} \mathscr{I}(R(X_x, X_y), R_c(X_x, X_y)) \quad (4)$$

where $\mathscr{I}$ represents an implicator, which is a mapping $\mathscr{I} : [0,1]^2 \rightarrow [0,1]$, decreasing in the first and increasing in the second argument, and for which $\mathscr{I}(0,0) = 1, \mathscr{I}(1,1) = 1, \mathscr{T}(0,1) = 1$ and $\mathscr{I}(1,0) = 0$. In this paper we will use the Łukasiewicz implicator, defined as follows: $\mathscr{I}(s,t) = \min(1, 1-s+t)$ for $s, t \in [0,1]$. The indiscernibility w.r.t. the decision class is defined as follows:

$$R_c(X_x, X_y) = \begin{cases} 1 & \text{if } X_{xc} = X_{yc} \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

The idea of the fuzzy-rough positive region is that instances on the border of a class (that is, for which there exists a similar instance in another class) will have a small membership value to POS_A compared to instances in the center of a class. This makes the fuzzy-rough positive region suitable to measure the quality of an instance as a typical representative of its class. Most applications of fuzzy rough sets for FS and PS data reduction are based on this approach (Jensen and Shen 2007, 2009; Cornelis et al. 2010; Jensen and Cornelis 2010).

2.4 Evolutionary algorithms for data reduction

Recently, the use of evolutionary algorithms in data reduction problems has become common in the machine learning field. This subsection surveys some interesting approaches for evolutionary FS and evolutionary PS.

In (Cano et al. 2003), a complete study on the use of evolutionary algorithms for prototype selection is carried out, highlighting four evolutionary methods to complete this task: CHC adaptive search algorithm (Eshelman 1991), steady-state genetic algorithm (SSGA) (Whitley 1989), generational genetic algorithm and population-based incremental learning. They concluded that the evolutionary algorithms selected outperform classical algorithms both in reduction rates and classification accuracy.

Other interesting evolutionary proposals for PS can be found in (García et al. 2008; Gil-Pita and Yao 2008; Ishibuchi and Nakashima 1998; Kuncheva 1995; Derrac et al. 2010a; García-Pedrajas et al. 2010). For a detailed survey on the field see (Derrac et al. 2010b).

Regarding FS, most of the evolutionary approaches are based on genetic algorithms, using both filter and wrapper approaches (Casillas et al. 2001; Gonzalez and Perez 2001; Oh et al. 2004; Rokach 2008). Another interesting proposal is (Inza et al. 2001), where an estimation of distribution algorithm based on bayesian networks is presented.

It is also possible to find evolutionary applications of simultaneous PS and FS. Both (Kuncheva and Jain 1999) and (Ishibuchi et al. 2001) propose a genetic algorithm to perform simultaneously the editing of the instance set and selection of the feature set. Another popular dual method is IGA (Ho et al. 2002), an intelligent genetic algorithm designed to tackle both PS and FS problems simultaneously, by the introduction of a special orthogonal crossover operator.

3 EFS-RPS: evolutionary feature selection for fuzzy rough set based prototype selection

In this section we describe the main components of EFS-RPS and its implementation details. The organization of this section follows a bottom-up order in which:

- Firstly, in Sect. 3.1, the PS algorithm based on fuzzy rough sets which forms the core of EFS-RPS is presented.
- Secondly, in Sect. 3.2, we describe the evolutionary algorithm developed for performing the FS process.
- Thirdly, in Sect. 3.3, we detail the way in which the subsets of features and prototypes obtained through the above methods are combined for preprocessing reference sets for the 1-NN classifier.
- Finally, in Sect. 4.4, a full description on the EFS-RPS is given, as a combination of all the components described before.

Input: The full set of instances TR
Output: A subset of prototypes RS
1 $CP = TR$ (Current Prototypes);
2 Compute $POS_A(x)$ for each $x \in CP$;
3 Sort prototypes in CP by their $POS_A(x)$ value in increasing order;
4 Compute nearest neighbors of TR in CP ;
5 bestAcc = 1-NNlooAccuracy(CP);
6 $RS = CP$;
7 continue = true ;
8 **while** CP *not empty* **do**
9 Mark the next prototype of RS, y, for deletion: $CP = CP - \{y\}$;
10 Update nearest neighbors of TR in CP ;
11 newAcc = 1-NNlooAccuracy(CP);
12 **if** $newAcc > bestAcc$ **then**
13 $RS = CP$;
14 bestAcc = newAcc;
15 **end**
16 **end**

Algorithm 1 A PS algorithm based on fuzzy rough set theory

3.1 A fuzzy rough set PS procedure

The quality of the instances in the training set can be assessed following the concepts of fuzzy rough set theory defined before and the definition of the fuzzy positive region of an instance $x \in X$. The membership of an instance to the positive region with respect to the current subset of features considered $B \subseteq M$ can serve as a noise measure for it.

In this fuzzy rough set PS procedure, instances are ordered with respect to their fuzzy positive region, in increasing order. Then, noisy instances are pruned iteratively, and the subsequently found prototype subsets are evaluated with respect to classification accuracy (accuracy is estimated by using a leave-one-out procedure with a 1-NN classifier, which we denote by 1-NNlooAccuracy). Instances with the same fuzzy positive region value are analyzed simultaneously, that is, marked for deletion at the same time. The best subset found is then identified as the subset of prototypes RS, which is the output of the algorithm. Algorithm 3 shows the pseudocode of this procedure.

Through the procedure, a list with the nearest neighbor of each prototype in TR is maintained. These nearest neighbors can only belong to the set RS; hence, every time a prototype from RS is removed, the list of neighbors is updated and the neighbor of every prototype of TR which is now missing is recomputed.

This neighbors list is used for estimating the 1-NN leave-one-out accuracy of the current RS set selected [through the 1-NNlooAccuracy (RS) procedure]. It helps the PS method to avoid the necessity of recomputing the nearest neighbors of each prototype in TR every time the procedure is used, thus saving computational resources.

Thanks to this optimization, the cost of the PS procedure can be computed by using the concept of partial evaluations. Throughout the FS and PS process of EFS-RPS, the computational resources spent are registered in the form of solutions evaluations. Every time a full classification of the TR is performed to estimate the accuracy of a solution, a full evaluation is spent.

In the specific case of this PS procedure, the complete cost is defined as follows:

- A full evaluation is spent the first time the nearest neighbor list is computed.
- Every time a neighbor has to be updated (because the old neighbor has been removed from RS) a partial evaluation is spent:

$$partialEvaluation = \frac{1}{\#\text{instances in TR}} \tag{6}$$

Therefore, the total cost of the PS algorithm can be redefined as

$$PSCost = 1 + \frac{\#\text{neighborsupdated}}{\#\text{instances in TR}} \tag{7}$$

This partial evaluations procedure, inspired by the one developed in (García et al. 2008), allows us to define a fair computational cost measure for the PS algorithm, which correctly represents the savings obtained through the use of the list of neighbors.

3.2 Searching features using an evolutionary algorithm

The second key element of EFS-RPS is its search method for selecting subsets of features. To accomplish this task, we have chosen SSGA as the evolutionary algorithm to perform the search.

A SSGA is a genetic algorithm in which only a reduced set of offspring is produced in each generation (two, in most cases). Parents are chosen 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. Algorithm 2 shows the pseudocode of SSGA.

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

Algorithm 2 SSGA pseudocode

The fitness function of our SSGA pursues a dual objective: The main task of the method is to search for subsets of features which increase the accuracy of the 1-NN classifier. However, a second task should be to reduce the size of the subsets selected, if this does not harm the accuracy rates obtained.

Hence, following the same set-up as in (Cano et al. 2003), where a similar approach is used in the core of evolutionary PS methods, for a given solution J (chromosome) of the SSGA, we define two variables:

- *AccRate*: The classification accuracy of a 1-NN classifier (1-NNAccuracy) when classifying the full training set using only the currently selected subset of features as a reference (and leave-one-out as validation scheme).

$$AccRate(J) = \text{1-NNAccuracy}(J) \quad (8)$$

- *RedRate* The rate of reduction achieved over the currently selected (maintained) features.

$$RedRate(J) = \frac{1.0 - \#\text{FeaturesSelected(J)}}{M} \quad (9)$$

Both variables are adjusted through a real-valued weighting factor α to equalize the strength of each term in the resulting fitness value. The final fitness function of the SSGA can be defined as

$$Fitness(J) = \alpha \cdot AccRate(J) + (1 - \alpha) \cdot RedRate(J) \quad (10)$$

The α value should be kept very high ($\alpha = 0.99$ turned out to be the best choice in our preliminary experiments) in order to avoid those cases in which an excessive deletion of features could be favored too much by the fitness function, resulting in a selection of an insufficient number of features for the final classification stage.

The configuration details of the SSGA are as follows:

- *Codification* The SSGA will use binary chromosomes to represent the solutions. Each bit will represent the state of each feature in the training set (**1** if the feature is selected; **0** if it is deleted).
- *Crossover operator* A two-point crossover operator has been considered. In each generation, this operator is applied twice, obtaining 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.
- *Selection of parents* A binary tournament procedure will be used to select parents in each generation.
- *Replacement strategy* The two worst individuals of the population are chosen for replacement, only if their fitness value is lower than the offspring's.

3.3 Simultaneous assessment of features and prototypes through the 1-NN classifier

While the search process performed by EFS-RPS is carried out, we will need to assess the quality of the solutions obtained. Mostly, this operation will consist of gathering two solutions (one representing a prototype subset, and another one representing a subset of features), combining them and estimating their quality through the 1-NN classifier.

Once both solutions have been gathered, their assessment is carried out by performing the following steps:

1. A copy of the training set is obtained and pruned, keeping only those prototypes indicated by the PS solution.
2. After the prototypes have been isolated, their features are also pruned, keeping this time only those indicated by the FS solution. The resulting subset is identified as the reference subset.
3. The 1-NN classifier is used to classify all the original training instances. This 1-NN classifier will use as reference set only the data preprocessed in the previous step.
4. The accuracy of this classification (that is, the ratio of training instances correctly classified over the total number of training instances), a value in [0, 1], will be used as the quality of the solutions.

This simple method allows us to evaluate subsets of features and prototypes within the general EFS-RPS framework. Since it involves the classification of the full training set, performing it has a cost associated of a full evaluation.

3.4 EFS-RPS global model

The EFS-RPS is composed by the fuzzy rough set based PS method and the SSGA FS algorithm defined before. Through the procedure for evaluating simultaneously feature and prototype subsets (defined in the previous subsection), EFS-RPS can merge the two search processes in an effective way, enabling the framework to obtain improved results from the existing synergy between the two basic data reduction methods on which it is based.

EFS-RPS begins by selecting a candidate set of prototypes, *bestPS*. This subset of the training set is used as reference for the SSGA, during a fixed number of evaluations (*Cycle Length*). Every time this limit is reached, the candidate set of prototypes is recomputed through the PS method, but considering only the best subset of features found so far, *bestFS*. If the new subset of prototypes computed is better than the previous one (in terms of leave-one-out accuracy), the latter is updated.

These processes are repeated until the algorithm is close to its end (at that point, no further update of the *bestPS* subset is allowed). Once the *Evaluations limit* is reached, the best subsets of prototypes and features found so far, *bestPS* and *bestFS* are used to prepare the final reference set, RS, which is obtained as the output of the algorithm. Algorithm 4 shows the EFS-RPS pseudocode.

```
   Input: A training set TR, Evaluations limit, Cycle Length
   Output: A reduced reference set RS
 1 bestFS=M;
 2 (bestAcc,bestPS)=PS-procedure(TR,bestFS);
 3 Initialize population of SSGA;
 4 optimizePS=true;
 5 cycle=0;
 6 while Evaluations limit not reached do
 7 |   bestFS=FSprocedureGeneration();
 8 |   if evaluations spent > (γ· Evaluations limit) then
 9 |   |   optimizePS=false;
10 |   end
11 |   if optimizePS AND cycle=Cycle Length then
12 |   |   //Current best features found are used in the
   |   |   evaluation of the solutions of the PS method;
13 |   |   (newAcc,newPS)=PSprocedure(TR,bestFS);
14 |   |   if newAcc > bestAcc then
15 |   |   |   bestAcc=newAcc;
16 |   |   |   bestPS=newPS;
17 |   |   end
18 |   |   cycle=0;
19 |   end
20 |   cycle+=2;
21 end
22 RS=reduceSet(TR,bestPS,bestFS);
```

Algorithm 3 The global EFS-RPS model

The main steps of the algorithm are detailed as follows:

- Instructions 1–2 extract a preliminary set of prototypes by applying the fuzzy rough PS procedure detailed in Sect. 3.1. In this first application of the PS procedure, all the features of the problem are considered.
- Instruction 3 initializes the population of the SSGA and its chromosomes are evaluated (using the procedure detailed in Sect. 3.3. The prototypes considered in this step are those stored in *bestPS*. Instructions 4 and 5 initialize *optimizePS* and *cycle* variables.
- Instruction 7 performs a full generation of the SSGA (again, the new chromosomes are evaluated using the procedure detailed in Sect. 3.3).
- Instructions 8 and 9 check if the end of the algorithm is close; that is, when more than $\gamma \cdot$ *Evaluations limit* evaluations have been spent (γ should be set close to 1, for example, $\gamma = 0.75$). If that limit is reached, the set of prototypes *bestPS* is no further optimized (see Instructions 11–20). Disabling these instructions in the last generations of the algorithms will help to improve the convergence capabilities of the SSGA.
- Instruction 10 checks if the set of prototypes *bestPS* has to be improved. If this phase is enabled (see Instructions 8 and 9) and *Cycle Length* evaluations have been spent since the last time this phase was carried out, the *bestPS* set can be improved.
- Instruction 13 generates a new candidate set of prototypes, *newPS* (only the features of the best subset found so far by the SSGA are considered in this case). The accuracy obtained by using this new set of prototypes is computed, *newAcc*. If it is higher than *bestAcc*, then the set *bestPS* is updated (Instructions 14–17).
- Instruction 22 gathers the best solutions found by the SSGA and the PS procedure, and creates a final reference set RS with those prototypes and features selected in the solutions.

The EFS-RPS algorithm loop is carried out until the specified limit of evaluations is reached. Then, the RS subset generated can be used as a reference set for the 1-NN classifier to classify new test instances.

4 Experimental study

This section describes the experimental study performed to test the performance of EFS-RPS:

- Section 4.1 lists the supervised classification problems considered and their main characteristics.
- Section 4.2 provides a description of the algorithms considered in the comparison and a definition of their parameter values.
- Section 4.3 describes the nonparametric statistical procedures considered for contrasting the results of the study.
- Section 4.4 shows the results obtained and analyzes them.

4.1 Data sets

We have selected a set of 38 classification data sets for our experimental study. These are well-known problems in the area, taken from the KEEL-dataset repository (Alcalá-Fdez et al. 2008, 2011)[2] and the UCI repository (Frank and Asuncion 2010). Table 1 summarizes their main characteristics. For each data set, we provide its number of instances (#Ins.), attributes (#At.) and classes (#Cl.).

The data sets considered are partitioned by using the 10-fold cross-validation (10-fcv) procedure (enabling us to follow a 5x10-fold cross-validation set-up in the study), and their values are normalized in the interval [0, 1] to equalize the influence of attributes with different range domains. In addition, instances with missing values have been discarded before the execution of the methods over the data sets.

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

Table 1 Description of the 38 data sets used in the study

Data set	#Ins.	#At.	#Cl.	Data set	#Ins.	#At.	#Cl.
Australian	690	14	2	Housevotes	435	16	2
Automobile	205	25	6	Iris	150	4	3
Balance	625	4	3	Led7Digit	500	7	10
Bands	539	19	2	Lymphography	148	18	4
Breast	286	9	2	Mammographic	961	5	2
Bupa	345	6	2	Monks	432	6	2
Car	1,728	6	4	New thyroid	215	5	3
Cleveland	303	13	5	Pima	768	8	2
Contraceptive	1,473	9	3	Saheart	462	9	2
Crx	690	15	2	Sonar	208	60	2
Dermatology	366	34	6	Spectfheart	267	44	2
Ecoli	336	7	8	Tae	151	5	3
Flare-solar	1,066	9	2	Tic-tac-toe	958	9	2
German	1,000	20	2	Vehicle	946	18	4
Glass	214	9	7	Vowel	990	13	11
Haberman	306	3	2	Wine	178	13	3
Hayes-Roth	160	4	3	Wisconsin	699	9	2
Heart	270	13	2	Yeast	1,484	8	10
Hepatitis	155	19	2	Zoo	101	16	7

4.2 Algorithms and parameter settings

In order to show the capabilities of EFS-RPS as a data preprocessor for 1-NN we have selected a representative set of comparison methods, including several evolutionary and fuzzy rough set based ones. The preprocessed training sets obtained as a result of the application of all these methods (including EFS-RPS) will be evaluated using a 1-NN classifier to classify the test (unseen) data. Euclidean distance will be considered in all the methods, whereas the overlap metric is considered for nominal attributes.

The comparison methods selected are the following:

- Evolutionary data reduction methods:
  - *FS-SSGA* A steady-state genetic algorithm for FS. This method follows the same design as the evolutionary component of EFS-RPS, but without including any kind of PS process. Hence, it is only focused on searching the best possible subset of features through a wrapper based evolutionary search.
  - *PS-SSGA* A steady-state genetic algorithm for PS. This method shares the same set-up as FS-SSGA, but it is focused on selecting prototypes, instead of features (it also uses binary chromosomes). Its objective is to find the most representative subset of prototypes from the training set through the evolutionary search process.
  - *FPS-SSGA* A steady-state genetic algorithm for simultaneous FS and PS. This method shares the same set-up as FS-SSGA and PS-SSGA, but both features and prototypes are encoded in the chromosomes. As output, this method will select a subset of features and a subset of prototypes, which are combined in the same way as the solutions of EFS-RPS.
- Fuzzy rough set data reduction methods:
  - *PS-FRW* A fuzzy rough set wrapper algorithm for PS. This method follows the same design as the PS component of EFS-RPS.
  - *FS-RST* A fuzzy rough set based feature selection method. It performs a heuristic search among the features of the training data, choosing the best ones according to how well they represent the full training set (using a measure of discernibility to evaluate the different subsets of features found). More details can be found in (Cornelis et al. 2010).
- Other algorithms:
  - *EIS-RFS* A hybrid data preprocessing method, which incorporates FS-RST and PS-SSGA for performing simultaneous FS and PS. A full description of this method can be found in (Derrac et al. 2012).
  - *1-NN* The 1-NN classifier is also included in the comparison. Its results, unmodified by any data preprocessing method, will give an insight of how well the rest of algorithms are improving the behavior of the base classifier, in terms of accuracy.

Table 2 Parameter specification for the algorithms tested in the experimentation

Algorithm	Parameters
EFS-RPS	Evaluations: 10,000, Population size: 50, Crossover probability: 1.0, Mutation probability: 0.005 per bit, α: 0.99 Cycle Length: 100, γ: 0.75
PS-SSGA	Evaluations: 10,000, Population size: 50, Crossover probability: 1.0, Mutation probability: 0.005 per bit, α: 0.5
FS-SSGA	Evaluations: 10,000, Population size: 50, Crossover probability: 1.0, Mutation probability: 0.005 per bit, α: 0.99
FPS-SSGA	Evaluations: 10,000, Population size: 50, Crossover probability: 1.0, Mutation probability: 0.005 per bit α (instances): 0.5, α (features): 0.99
PS-FRW	–
FS-RST	*MaxGamma*: 1.0
EIS-RFS	Evaluations: 10,000, Population size: 50, Crossover probability: 1.0, Mutation probability: 0.005 per bit, α: 0.5 *MaxGamma*: 1.0, UpdateFS: 100, β: 0.75
1-NN	–

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, which will be applied for all the data sets. Table 2 summarizes them.

Essentially, 10,000 evaluations are allowed for every method. SSGA base parameters are set-up to a classical configuration, and the α value in the fitness function is set to 0.5 if the reduction rate is computed over instances [as recommended in Cano et al. (2003)] and to 0.99 if it is computed over features [as recommended in Derrac et al. (2010a)]. The rest of parameters are set to the values recommended by the authors of each technique.

4.3 Statistical procedures

We have considered the use of hypothesis testing techniques to provide statistical support for the analysis of the results of the experimental study. Concretely, we will use nonparametric tests (Sheskin 2011), since the initial conditions that guarantee the reliability of the parametric tests (independence, normality and homocedasticity) may not be satisfied, causing the statistical analysis to lose credibility (García and Herrera 2008; García et al. 2009).

Throughout the study, we perform several multiple comparisons between the algorithms considered. To do so, we will use the Friedman test in order to detect statistical differences among a group of results. A second property of this test is that the ranks computed for obtaining the Friedman statistic can be also considered to sort the algorithm by its relative performance (where the lower the rank obtained, the better the performance of the algorithm). The process followed to compute the final ranks is as follows:

1. Gather observed results for each pair algorithm/data set (for example, average the results obtained after the cross-validation process).
2. For each data set, rank the values from 1 (best result) to n (worst result), where n is the number of algorithms considered in the comparison. If ties appear, assign midranks.
3. Average the ranks obtained in all data sets to obtain the final rank.

After computing the ranks, if the p-value of the Friedman test is significantly low (at a 0.05 level of significance), the existence of significant differences between the algorithms evaluated is assumed. From this point, a control algorithm can be chosen (the one with the lowest rank, that is, the best performing one), and post-hoc procedures (in our case the Holm and Finner procedures (García et al. 2010)) can be applied to determine which algorithms are significantly outperformed by the control one.

More 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*.[3]

4.4 Results and analysis

In this subsection we report the results obtained in the full experimental study. Table 3 shows the average accuracy results obtained in the test phase (considering a 5x10-fold cross-validation set-up, that is, averaging the results of five independent schemes of 10-fold cross-validation). For each algorithm and data set, the average accuracy and standard deviation are provided. The best result in each data set is highlighted in bold. Moreover, the table also provides average results over all data sets and the number of times that each algorithm obtains the best result for a single data set.

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

Table 3 Accuracy results in test phase

Data set	EFS-RPS	FS-SSGA	PS-SSGA	FPS-SSGA	PS-FRW	FS-RST	EIS-RFS	1-NN
Australian	85.12 ± 4.53	85.07 ± 3.49	85.65 ± 2.77	85.36 ± 3.31	84.64 ± 3.15	81.45 ± 4.52	**85.66** ± 2.27	81.45 ± 4.29
Automobile	**82.05** ± 8.55	79.61 ± 6.87	63.78 ± 14.84	69.38 ± 6.33	76.07 ± 7.48	78.97 ± 10.32	61.17 ± 11.58	77.93 ± 6.68
Balance	85.04 ± 6.81	70.89 ± 9.80	86.40 ± 3.08	84.31 ± 4.85	**90.56** ± 1.57	79.04 ± 6.81	85.92 ± 2.62	79.04 ± 6.46
Bands	**74.42** ± 5.45	72.35 ± 6.70	69.77 ± 9.22	64.95 ± 8.69	72.01 ± 9.66	66.61 ± 6.32	64.57 ± 5.91	74.04 ± 6.94
Breast	69.94 ± 7.01	70.98 ± 5.72	70.94 ± 4.63	**73.42** ± 8.34	66.03 ± 6.96	60.97 ± 10.44	69.29 ± 5.73	65.35 ± 6.39
Bupa	63.98 ± 6.82	59.91 ± 10.19	61.14 ± 9.37	62.72 ± 8.40	59.59 ± 9.97	62.51 ± 7.78	**65.72** ± 8.79	61.08 ± 6.88
Car	90.68 ± 1.31	90.68 ± 1.51	89.29 ± 2.55	**93.34** ± 1.37	85.65 ± 3.03	70.02 ± 0.17	91.67 ± 3.31	85.65 ± 1.91
Cleveland	53.81 ± 7.66	51.47 ± 9.47	52.82 ± 4.47	56.13 ± 6.07	**56.82** ± 8.87	52.51 ± 9.49	55.16 ± 5.82	53.14 ± 7.45
Contraceptive	43.31 ± 4.21	41.96 ± 3.57	44.54 ± 4.61	45.15 ± 2.32	44.47 ± 2.57	42.63 ± 3.73	**45.42** ± 5.14	42.77 ± 3.69
Crx	**85.07** ± 3.75	81.16 ± 7.61	84.64 ± 4.22	84.64 ± 5.08	83.04 ± 4.64	81.30 ± 6.28	84.93 ± 5.72	79.57 ± 5.12
Dermatology	95.65 ± 4.07	**96.71** ± 2.85	94.84 ± 4.66	95.36 ± 3.83	95.92 ± 3.43	91.59 ± 3.69	94.81 ± 4.18	95.35 ± 3.64
Ecoli	79.79 ± 7.51	78.90 ± 7.30	80.38 ± 5.69	77.70 ± 5.52	**83.33** ± 6.13	76.58 ± 14.73	82.14 ± 8.42	80.70 ± 7.51
Flare-solar	63.61 ± 3.12	62.76 ± 3.65	64.82 ± 3.37	**67.35** ± 4.12	66.70 ± 3.61	63.23 ± 5.56	66.32 ± 2.94	55.54 ± 3.20
German	**72.00** ± 3.40	69.50 ± 2.68	70.40 ± 3.24	70.10 ± 3.48	70.30 ± 4.57	67.90 ± 3.41	70.80 ± 4.24	70.50 ± 4.25
Glass	71.52 ± 14.45	71.80 ± 14.30	67.10 ± 14.74	71.23 ± 10.64	73.20 ± 14.31	**74.50** ± 13.17	67.35 ± 11.83	73.61 ± 11.91
Haberman	72.81 ± 5.62	72.81 ± 6.15	71.23 ± 5.40	**72.83** ± 5.99	67.65 ± 4.73	65.68 ± 6.58	71.56 ± 7.34	66.97 ± 5.46
Hayes-Roth	**83.93** ± 9.03	83.93 ± 8.33	69.15 ± 11.69	79.80 ± 11.65	75.46 ± 10.29	76.07 ± 14.07	80.86 ± 11.70	35.70 ± 9.11
Heart	78.89 ± 6.77	76.67 ± 6.06	81.11 ± 7.90	**82.59** ± 6.31	82.22 ± 5.18	78.89 ± 6.77	80.74 ± 6.34	77.04 ± 8.89
Hepatitis	81.92 ± 10.03	76.21 ± 7.89	79.33 ± 8.71	80.67 ± 6.13	82.04 ± 10.26	79.50 ± 7.95	**82.58** ± 7.99	82.04 ± 11.09
Housevotes	**96.31** ± 3.65	94.01 ± 4.53	93.79 ± 3.43	94.46 ± 4.37	92.38 ± 5.79	90.78 ± 6.47	94.48 ± 3.67	91.24 ± 5.41
Iris	**96.00** ± 4.66	95.33 ± 4.50	94.67 ± 2.81	94.67 ± 4.22	95.33 ± 5.49	93.33 ± 5.44	**96.00** ± 4.92	93.33 ± 5.16
Led7Digit	63.00 ± 7.54	63.00 ± 6.94	**73.40** ± 2.84	71.40 ± 4.81	63.20 ± 3.43	63.60 ± 5.87	73.20 ± 4.99	40.20 ± 9.48
Lymphography	75.21 ± 9.75	**78.49** ± 9.12	77.92 ± 9.39	74.92 ± 10.79	73.87 ± 9.17	77.38 ± 11.21	77.15 ± 12.15	73.87 ± 8.77
Mammographic	79.42 ± 4.26	75.86 ± 6.07	79.50 ± 3.85	80.15 ± 6.23	79.09 ± 3.80	75.76 ± 4.97	**80.65** ± 4.51	76.38 ± 5.67
Monks	**100.00** ± 0.00	**100.00** ± 0.00	83.53 ± 6.21	98.64 ± 3.07	77.70 ± 5.37	77.91 ± 5.71	**100.00** ± 0.00	77.91 ± 5.42
New thyroid	96.75 ± 2.24	96.30 ± 1.95	**98.16** ± 3.20	96.32 ± 3.60	96.73 ± 3.18	97.23 ± 2.39	96.77 ± 4.83	97.23 ± 2.26
Pima	73.35 ± 5.21	67.70 ± 4.59	72.26 ± 4.44	73.83 ± 3.15	74.49 ± 3.49	70.33 ± 3.71	**74.80** ± 3.71	70.33 ± 3.53
Saheart	69.05 ± 6.69	61.24 ± 3.91	69.27 ± 3.70	67.99 ± 5.69	**71.66** ± 6.12	64.49 ± 4.21	68.82 ± 7.16	64.49 ± 3.99
Sonar	**89.43** ± 6.65	84.62 ± 8.65	75.45 ± 11.74	75.50 ± 12.59	85.57 ± 7.14	81.69 ± 9.83	80.76 ± 7.88	85.55 ± 7.51
Spectfheart	74.56 ± 8.79	74.17 ± 6.34	75.31 ± 5.96	75.34 ± 7.31	**78.36** ± 7.22	70.04 ± 8.00	76.82 ± 7.07	69.70 ± 6.55
Tae	**62.38** ± 13.09	62.37 ± 14.17	54.42 ± 11.63	55.62 ± 13.70	61.08 ± 15.09	60.42 ± 14.29	52.08 ± 11.22	40.50 ± 8.89
Tic-tac-toe	83.20 ± 3.25	**83.51** ± 3.10	78.71 ± 3.36	77.87 ± 5.25	73.07 ± 2.28	73.07 ± 2.70	78.29 ± 5.07	73.07 ± 2.56
Vehicle	**72.70** ± 5.40	70.58 ± 4.92	66.91 ± 4.38	70.92 ± 3.84	68.20 ± 5.65	65.56 ± 6.14	65.37 ± 6.71	70.10 ± 5.90
Vowel	99.19 ± 0.90	99.19 ± 0.80	91.62 ± 3.01	89.60 ± 3.96	99.09 ± 1.00	91.58 ± 4.29	98.81 ± 2.10	**99.39** ± 0.85
Wine	95.52 ± 3.53	94.90 ± 3.30	92.68 ± 7.91	94.93 ± 3.17	95.52 ± 6.84	95.49 ± 4.40	**97.19** ± 5.09	95.52 ± 4.85
Wisconsin	95.85 ± 1.25	95.14 ± 2.62	96.13 ± 2.95	95.86 ± 2.47	**96.86** ± 2.41	95.57 ± 2.73	96.42 ± 1.55	95.57 ± 2.59
Yeast	**54.23** ± 4.01	52.30 ± 3.94	54.18 ± 4.38	53.50 ± 3.77	52.90 ± 3.62	52.23 ± 4.39	53.37 ± 3.36	50.47 ± 3.91
Zoo	**98.33** ± 3.60	95.42 ± 6.00	94.22 ± 7.94	90.72 ± 7.09	98.33 ± 3.60	96.50 ± 4.61	96.39 ± 4.80	92.81 ± 6.57
Average	**79.16** ± 5.65	77.30 ± 5.78	76.56 ± 6.01	77.61 ± 5.83	77.61 ± 5.82	74.81 ± 6.66	78.00 ± 5.86	73.56 ± 5.81
Best result (of 38)	13	4	2	5	6	1	9	1

Table 4 Results of Friedman, Holm and Finner tests

Algorithm	Friedman ranking	Holm p-value	Finner p-value
FS-RST	7.4211	**0.000002**	**0.000002**
1-NN	7.0263	**0.000030**	**0.000017**
FS-SSGA	6.2368	**0.003437**	**0.001472**
PS-SSGA	5.4605	**0.089441**	**0.031968**
FPS-SSGA	5.1053	0.242954	**0.089710**
PS-FRW	5.0132	0.242954	**0.094809**
EIS-RFS	4.2368	0.544390	0.544390
EFS-RPS	**3.8158**	–	–
Friedman p-value		$<10^{-6}$	

Table 5 Average reduction results over features and instances

Data set	Features					Instances				
	EFS-RPS	FS-SSGA	FPS-SSGA	FS-RST	EIS-RFS	EFS-RPS	PS-SSGA	FPS-SSGA	PS-FRW	EIS-RFS
Australian	0.7500	0.8071	0.7929	0.0000	0.1571	0.4288	0.8799	0.8808	0.4032	0.8872
Automobile	0.6760	0.7560	0.7160	0.3265	0.3560	0.0727	0.8444	0.8309	0.0640	0.8531
Bal	0.0000	0.3000	0.0000	0.0000	0.0000	0.5329	0.8686	0.8085	0.5846	0.8464
Bands	0.4842	0.6526	0.5526	0.3750	0.4263	0.3153	0.8472	0.8363	0.3267	0.8689
Bre	0.5444	0.6667	0.7111	0.2111	0.2111	0.0454	0.9441	0.9779	0.0408	0.9476
Bupa	0.4000	0.3667	0.4333	0.1274	0.0000	0.3838	0.8644	0.8644	0.3316	0.8502
Car	0.1667	0.1667	0.1667	0.1667	0.1667	0.0000	0.7681	0.7592	0.0000	0.8279
Cleveland	0.5462	0.7385	0.6077	0.3908	0.0462	0.7141	0.9171	0.9289	0.7352	0.9014
Contraceptive	0.4444	0.4556	0.5889	0.0360	0.0667	0.4133	0.7530	0.7530	0.4463	0.7637
Crx	0.4733	0.5667	0.5533	0.2000	0.1800	0.1685	0.8816	0.8805	0.1750	0.8914
Dermatology	0.5500	0.6676	0.4735	0.4354	0.3854	0.1520	0.9448	0.9414	0.1332	0.9502
Ecoli	0.1571	0.1714	0.1857	0.2286	0.1286	0.3179	0.9077	0.9130	0.3201	0.8882
Flare-solar	0.4778	0.5111	0.5778	0.1556	0.0556	0.8093	0.8391	0.8005	0.7963	0.8122
German	0.6800	0.5150	0.7450	0.1450	0.2350	0.0423	0.7914	0.7928	0.0022	0.8014
Glass	0.4222	0.4444	0.4556	0.0168	0.0444	0.1322	0.8791	0.8791	0.1422	0.8718
Haberman	0.6667	0.6667	0.5333	0.0254	0.0000	0.6560	0.9379	0.9379	0.6385	0.9306
Hayes-Roth	0.2500	0.2500	0.2500	0.1000	0.2500	0.2343	0.8384	0.8452	0.2172	0.8544
Heart	0.6231	0.4538	0.5692	0.1846	0.2308	0.6523	0.9506	0.9230	0.6617	0.9255
Hepatitis	0.6000	0.6684	0.5421	0.4263	0.5368	0.0471	0.9226	0.9355	0.0229	0.9262
Housevotes	0.6313	0.7000	0.7313	0.0188	0.3500	0.0389	0.9410	0.9653	0.0128	0.9387
Iris	0.4000	0.4000	0.4500	0.0000	0.1250	0.3415	0.9481	0.9481	0.1978	0.9511
Led7Digit	0.0143	0.0143	0.0000	0.0143	0.0000	0.5122	0.9071	0.9491	0.5122	0.9416
Lym	0.5111	0.6500	0.6500	0.2611	0.4444	0.0000	0.8994	0.9234	0.0000	0.9257
Mammographic	0.6600	0.5000	0.6200	0.3396	0.0000	0.4644	0.8229	0.7829	0.4722	0.8322
Monks	0.5000	0.5000	0.5333	0.0000	0.5000	0.3922	0.8570	0.9406	0.3760	0.9342
New thyroid	0.3200	0.3000	0.3800	0.0000	0.0600	0.6481	0.9571	0.9571	0.6652	0.9473
Pima	0.5500	0.5750	0.4375	0.0000	0.0000	0.5382	0.8187	0.8187	0.5298	0.7911
Saheart	0.4556	0.6333	0.5778	0.0000	0.0000	0.7245	0.8841	0.8778	0.6857	0.8668
Sonar	0.5767	0.6633	0.6600	0.7183	0.2900	0.0422	0.8595	0.8974	0.0283	0.8899
Spectfheart	0.4977	0.6750	0.6614	0.2750	0.2727	0.5428	0.9426	0.9409	0.5331	0.9497
Tae	0.3000	0.4000	0.2200	0.1183	0.1291	0.1902	0.8727	0.8992	0.1847	0.8764
Tic-tac-toe	0.2444	0.2444	0.2889	0.0000	0.0000	0.0000	0.7917	0.8047	0.0000	0.8655
Vehicle	0.4722	0.4833	0.4778	0.2944	0.2549	0.3256	0.7895	0.7927	0.3082	0.8211
Vowel	0.3077	0.3077	0.3538	0.2894	0.2640	0.0162	0.7201	0.7366	0.0091	0.7552
Wine	0.4846	0.4538	0.4538	0.5231	0.3308	0.6593	0.9538	0.9557	0.5905	0.9451
Wisconsin	0.4556	0.3889	0.3222	0.0000	0.0444	0.3672	0.9027	0.9048	0.4036	0.9103
Yeast	0.1000	0.0875	0.1625	0.1256	0.0375	0.4428	0.7485	0.7485	0.4272	0.7550
Zoo	0.7063	0.7125	0.3750	0.2750	0.2125	0.2349	0.8714	0.8468	0.2150	0.8634
Average	0.4500	0.4872	0.4687	0.1791	0.1787	0.3316	0.8702	0.8731	0.3209	0.8779

Table 4 summarizes the results of the Friedman test, and the post-hoc procedures (Holm and Finner), performed to contrast the results obtained concerning classification accuracy. Average rankings and p-value are reported for the Friedman test, and the best (lowest) rank is highlighted in bold. Regarding the post-hoc methods, adjusted p-values are provided, highlighting in bold those which represent significant differences (at a 0.1 level of significance).

Table 5 shows the average reduction rates achieved through the application of every data reduction method. On the left-hand side it shows the reduction achieved over the set of features (that is, the ratio of features selected over the original number of features of the problem) and, on the

Table 6 Average time elapsed (training phase), in seconds

Data set	EFS-RPS	FS-SSGA	PS-SSGA	FPS-SSGA	PS-FRW	FS-RST	EIS-RFS
Australian	111.35	161.39	79.16	48.14	21.53	0.70	82.54
Automobile	16.80	50.27	14.40	8.55	0.40	0.36	30.48
Bal	29.27	88.56	38.71	38.33	0.21	0.03	54.44
Bands	85.28	287.05	75.96	63.93	13.09	1.41	116.68
Bre	13.77	43.73	8.86	5.67	0.13	0.09	12.32
Bupa	13.60	32.65	13.70	11.33	1.25	0.04	20.71
Car	475.93	1619.67	442.19	520.30	1.17	0.20	560.60
Cleveland	20.25	33.37	11.89	9.04	1.61	0.10	19.29
Contraceptive	352.47	704.06	348.66	306.52	9.09	1.32	316.30
Crx	114.32	220.59	79.72	70.56	6.73	0.46	86.38
Dermatology	66.46	186.50	35.90	27.80	2.33	0.30	60.20
Ecoli	14.63	37.50	10.92	11.17	1.36	0.05	20.68
Flare-solar	183.55	349.09	160.00	123.76	0.95	0.01	183.44
German	448.56	591.00	252.59	167.51	1.63	2.07	304.94
Glass	6.95	15.93	5.39	5.18	0.53	0.05	10.30
Haberman	7.39	13.63	7.09	6.06	0.22	0.01	9.41
Hayes-Roth	1.79	5.00	2.68	2.52	0.09	0.02	3.86
Heart	15.19	32.98	8.03	7.01	1.18	0.06	14.57
Hepatitis	7.89	13.08	3.83	3.21	0.11	0.04	8.50
Housevotes	50.86	82.91	24.98	17.38	0.31	0.02	39.42
Iris	1.64	5.22	2.44	2.33	0.17	0.02	4.40
Led7Digit	36.43	88.31	25.05	28.87	0.18	0.01	40.50
Lym	6.44	11.77	3.97	3.23	0.09	0.02	8.14
Mammographic	123.48	205.34	116.67	77.57	1.62	0.20	127.75
Monks	18.91	46.18	20.72	14.22	0.16	0.02	27.92
New thyroid	3.93	11.76	3.68	3.63	0.40	0.01	8.03
Pima	88.43	175.95	85.38	72.09	17.65	0.29	96.68
Saheart	34.33	62.64	25.98	22.45	4.25	0.15	33.45
Sonar	39.49	66.79	17.12	13.86	0.78	0.40	136.71
Spectfheart	38.13	80.53	15.14	15.25	3.98	0.27	40.33
Tae	2.48	9.41	2.42	2.37	0.11	0.03	3.27
Tic-tac-toe	144.63	348.57	150.31	132.37	0.59	0.06	176.75
Vehicle	409.15	671.39	220.63	183.52	47.11	6.13	495.09
Vowel	495.70	867.64	270.26	241.58	60.75	4.39	461.71
Wine	3.77	14.80	3.62	3.34	0.48	0.03	7.67
Wisconsin	70.02	159.05	55.66	50.62	2.81	0.10	73.61
Yeast	295.60	825.97	354.55	364.96	109.57	1.47	420.58
Zoo	2.43	5.12	2.52	2.46	0.05	0.01	5.31
Average	101.35	216.46	78.97	70.75	8.28	0.55	108.50

right-hand side, the reduction achieved over the set of instances (the ratio of prototypes selected over the total number of instances of the original training set).

Finally, Table 6 reports the time elapsed for the methods in training phase (in seconds).[4] Note that the running times in the test phase are not reported since they are too low to show interesting differences. Furthermore, the efficiency in the test phase is already reflected by the reduction rates achieved (the higher the reduction rates are, the less running time will be needed).

We can draw the following conclusions:

- The new hybrid approach, EFS-RPS, obtains the best results in accuracy. As Table 3 shows, it achieves the

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

best average result, and the best result for 13 out of 38 data sets. The introduction of fuzzy rough set theory to improve the evolutionary techniques turns out to be very effective when its results are compared with the basic techniques considered in isolation.

- All the data reduction methods considered in the study outperform the basic 1-NN accuracy, PS-FRW being the best non-hybrid technique when the average ranking is considered. FS and PS, when performed in an effective way—such as using the preprocessing methods considered in the study—are able to improve the accuracy of the 1-NN classifier (sometimes providing a substantial improvement, such as in the case of EFS-RPS, with an average accuracy increase of more than 5 %. This aspect is further reflected by the starplot represented in Fig. 1).
- The hybridization performed to design EFS-RPS has not damaged the reduction power of the base techniques on which it is based: as Table 5 reports, its reduction rate over features is similar to the one obtained by FS-SSGA, the evolutionary FS method which guides its search; its reduction rate over instances is also similar to the one obtained by PS-FRW, its basic PS inner procedure. Hence, the hybridization has shown to be effective for increasing the accuracy of the preprocessing technique without damaging the reduction capabilities of the standalone procedures.
- Concerning running time, Table 6 shows that EFS-RPS has an average behavior: it is somewhat slower than the evolutionary methods with high prototype reduction power (PS-SSGA, FPS-SSGA), but faster than the evolutionary methods that focus only on performing FS (FS-SSGA). It is also faster than the hybrid preprocessing method considered in the comparison (EIS-RFS).

The statistical study, performed to contrast the results obtained in accuracy, confirms our analysis: the Friedman

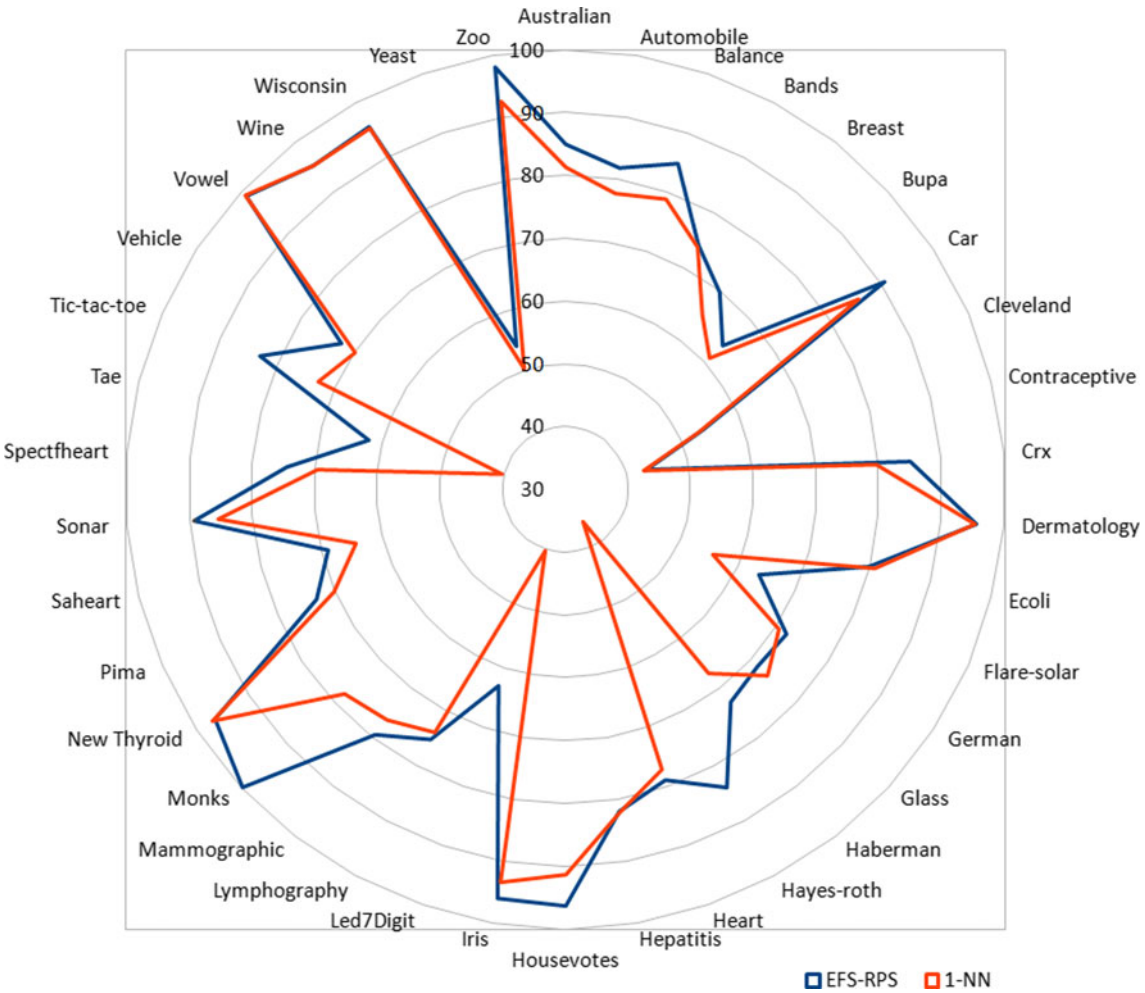

Fig. 1 Starplot depicting the enhancement in accuracy of the 1-NN classifier when EFS-RPS is used for preprocessing the data. The differences between the areas of the star represent absolute differences between the precision of both classifiers classifying unseen instances in test phase

test detects significant differences among the methods (with a p-value $<10^{-6}$) and highlights EFS-RPS as the best performing one, with a rank of 3.8158 (the lowest; see Table 4). The Holm test establishes the existence of significant differences between EFS-RPS and FS-RST, 1-NN, FS-SSGA and PS-SSGA, which are further expanded by the Finner test, marking as significant the comparisons between EFS-RPS and FPS-SSGA and PS-FRW. Only the differences between EFS-RPS and EIS-RFS are not strong enough to be marked as significant.

Differences between the behavior of EIS-RFS and EFS-RPS can be found if a smaller subset of problems is considered (instead of the large set of domains chosen in this study); thus, a suitable choice may depend upon the specific problem to tackle. However, we can point out that the small difference found between both hybrid methods can be caused due to the lower variation in the possible feature sets selected by EIS-RPS, which is not an issue for the evolutionary FS part of EFS-RPS.

In summary, EFS-RPS can be highlighted for being a highly accurate method for performing dual data reduction (including FS and PS) for the 1-NN classifier. It improves the accuracy of evolutionary and fuzzy rough set based data reduction approaches without losing reduction power and without increasing the time complexity of the procedures considered. Therefore, it is a competent method for performing data reduction which can be applied for improving the performance of the 1-NN in any standard supervised classification domain.

5 Conclusions

In this work, we have proposed a new approach based on fuzzy rough sets and evolutionary algorithms for performing a simultaneous process of FS and PS. This data reduction process is specifically designed to improve the performance of the 1-NN classifier, both regarding test accuracy and computational complexity.

The results achieved by EFS-RPS in the experimental study performed have shown that it offers the best results among all the related techniques selected for the comparison; that is, the hybrid approach outperforms those methods based only in either evolutionary techniques or fuzzy rough set ones. Nonparametric statistical procedures have been used to contrast this results, supporting the conclusions arrived at.

These promising results allow us to point out further extensions of the EFS-RPS model, and new directions of research related. One of them would be to test the behavior of the model when considering other base classifiers. That is, to apply the model and preprocess the data using other classifiers different than 1-NN. This extension would put our approach in the field of Training Set Selection (see (Kim 2006; Cano et al. 2007, 2008) for some promising applications, and (Derrac et al. 2010b; García-Pedrajas 2011) for two reviews on evolutionary approaches to the field), analogous to PS, providing more generality in the range of domains in which EFS-RPS can be applied.

Another interesting trend of research can be focused on particular traits of the data. Imbalanced data sets (He and Garcia 2009) pose a problem nowadays in many applications of research. This tough problem requires the definition of specific methods, measures and evaluation procedures; however, the application of evolutionary preprocessing methods for nearest neighbor classifiers and rough set theory with success is still possible (see García and Herrera 2009; Ramentol et al. 2012, respectively). Hence, further research on extensions of EFS-RPS could be focused on obtaining a new version of the model, suitable for tackling imbalanced domains.

Supported by the Spanish Ministry of Science and Technology (Project TIN2011-28488) and by the Regional Project P10-TIC-06858. J. Derrac holds an FPU scholarship from the Spanish Ministry of Education. C. Cornelis holds a Ramón y Cajal research contract from the Spanish Ministry of Science and Innovation.

References

Aha DW, Kibler D, Albert MK (1991) Instance-based learning algorithms. Mach Learn 6:37–66

Alcalá-Fdez J, Sánchez L, García S, del Jesus MJ, Ventura S, Garrell JM, Otero J, Romero C, Bacardit J, Rivas VM, Fernández JC, Herrera F (2008) KEEL: a software tool to assess evolutionary algorithms for data mining problems. Soft Comput 13(3):307–318

Alcalá-Fdez J, Fernández A, Luengo J, Derrac J, García S, Sánchez L, Herrera F (2011) Keel data-mining software tool: data set repository, integration of algorithms and experimental analysis framework. J Mult Valued Log Soft Comput 17(2–3):255–287

Almuallim H, Dietterich T (1991) Learning with many irrelevant features. In: Proceedings of the 9th national conference on artificial intelligence, vol 2, Anaheim, CA, USA, July 14–19, The MIT Press, pp 547–552

Alpaydin E (2010) Introduction to machine learning, 2nd edn. The MIT Press, Cambridge

Bell G, Hey T, Szalay A (2009) Beyond the data deluge. Science 323:1297–1298

Cano JR, Herrera F, Lozano M (2003) Using evolutionary algorithms as instance selection for data reduction in KDD: An experimental study. IEEE Trans Evol Comput 7(6):561–575

Cano JR, Herrera F, Lozano M (2007) Evolutionary stratified training set selection for extracting classification rules with trade-off precision-interpretability. Data Knowl Eng 60:90–100

Cano JR, Herrera F, Lozano M, García S (2008) Making CN2-SD subgroup discovery algorithm scalable to large size data sets using instance selection. Expert Syst Appl 35:1949–1965

Casillas J, Cordon O, Del Jesus MJ, Herrera F (2001) Genetic feature selection in a fuzzy rule-based classification system learning process for high-dimensional problems. Inf Sci 136:135–157

Chen Y, Garcia EK, Gupta MR, Rahimi A, Cazzanti L (2009) Similarity-based classification: concepts and algorithms. J Mach Learn Res 10:747–776

Cornelis C, Jensen R, Hurtado G, Slezak D (2010) Attribute selection with fuzzy decision reducts. Inf Sci 180:209–224

Cover TM, Hart PE (1967) Nearest neighbor pattern classification. IEEE Trans Inf Theory 13(1):21–27

De Cock M, Cornelis C, Kerre EE (2007) Fuzzy rough sets: The forgotten step. IEEE Trans Fuzzy Syst 15(1):121–130

Derrac J, García S, Herrera F (2010a) IFS-CoCo: Instance and feature selection based on cooperative coevolution with nearest neighbor rule. Pattern Recognit 43(6):2082–2105

Derrac J, García S, Herrera F (2010b) A survey on evolutionary instance selection and generation. Int J Appl Metaheur Comput 1(1):60–92

Derrac J, Cornelis C, García S, Herrera F (2012) Enhancing evolutionary instance selection algorithms by means of fuzzy rough set based feature selection. Inf Sci 186(1):73–92

Destercke S (2012) A k-nearest neighbours method based on imprecise probabilities. Soft Comput 16(5):833–844

Dubois D, Prade H (1990) Rough fuzzy sets and fuzzy rough sets. Int J General Syst 17:191–209

Eiben AE, Smith JE (2003) Introduction to Evolutionary Computing. Natural Computing, Springer-Verlag, Berlin

Eshelman LJ (1991) The CHC adaptive search algorithm: how to have safe search when engaging in nontraditional genetic recombination. In: Rawlins GJE (ed) Foundations of genetic algorithms, Morgan Kaufmann, San Mateo, pp 265–283

Ferrandiz S, Boullé M (2010) Bayesian instance selection for the nearest neighbor rule. Mach Learn 81(81):229–256

Franco A, Maltoni D, Nanni L (2010) Data pre-processing through reward-punishment editing. Pattern Anal Appl 13:367–381

Frank A, Asuncion A (2010) UCI machine learning repository. http://archive.ics.uci.edu/ml

Freitas AA (2002) Data mining and knowledge discovery with evolutionary algorithms. Springer-Verlag, Berlin

García S, Herrera F (2008) An extension on statistical comparisons of classifiers over multiple data sets for all pairwise comparisons. J Mach Learn Res 9:2677–2694

García S, Herrera F (2009) Evolutionary undersampling for classification with imbalanced datasets: Proposals and taxonomy. Evol Comput 17(3):275–306

García S, Cano JR, Herrera F (2008) A memetic algorithm for evolutionary prototype selection: A scaling up approach. Pattern Recognit 41(8):2693–2709

García S, Fernández A, Luengo J, Herrera F (2009) A study of statistical techniques and performance measures for genetics-based machine learning: accuracy and interpretability. Soft Comput 13(10):959–977

García S, Fernández A, Luengo J, Herrera F (2010) Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: experimental analysis of power. Inf Sci 180:2044–2064

García S, Derrac J, Cano JR, Herrera F (2012a) Prototype selection for nearest neighbor classification: taxonomy and empirical study. IEEE Trans Pattern Anal Mach Intell 34(3):417–435

García S, Luengo J, Sáez JA, López V, Herrera F (2012b) A survey of discretization techniques: taxonomy and empirical analysis in supervised learning. IEEE Trans Knowl Data Eng (in press)

García-Pedrajas N (2011) Evolutionary computation for training set selection. Wiley Interdiscip Rev Data Min Knowl Dis 1(6):512–523

García-Pedrajas N, Romero JA, Ortiz-Boyer D (2010) A cooperative coevolutionary algorithm for instance selection for instance-based learning. Mach Learn 78:381–420

Ghosh A, Jain LC (eds) (2005) Evolutionary computation in data mining. Springer-Verlag, Berlin

Gil-Pita R, Yao X (2008) Evolving edited k-nearest neighbor classifiers. Int J Neural Syst 18(6):1–9

Gonzalez A, Perez R (2001) Selection of relevant features in a fuzzy genetic learning algorithm. IEEE Trans Syst Man Cybern 31(3): 417–425

Guyon I, Elisseeff A (2003) An introduction to variable and feature selection. J Mach Learn Res 3:1157–1182

Guyon I, Gunn S, Nikravesh M, Zadeh LA (eds) (2006) Feature extraction: foundations and applications. Springer, Berlin

Hart PE (1968) The condensed nearest neighbour rule. IEEE Trans Inf Theory 18(5):515–516

He H, Garcia EA (2009) Learning from imbalanced data. IEEE Trans Knowl Data Eng 21:1263–1284

He Q, Wu C (2011) Membership evaluation and feature selection for fuzzy support vector machine based on fuzzy rough sets. Soft Comput 15(6):1105–1114

Ho SY, Liu CC, Liu S (2002) Design of an optimal nearest neighbor classifier using an intelligent genetic algorithm. Pattern Recognit Lett 23(13):1495–1503

Inza I, Larrañaga P, Sierra B (2001) Feature subset selection by bayesian networks: a comparison with genetic and sequential algorithms. Int J Approx Reason 27:143–164

Ishibuchi H, Nakashima T (1998) 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). Lecture notes in computer science, vol 1585, pp 82–89

Ishibuchi H, Nakashima T, Nii M (2001) Genetic-algorithm-based instance and feature selection. In: Liu H, Motoda H (eds) Instance selection and construction for data mining, Kluwer Academic Publishers, Dordrecht, pp 95–112

Jensen R, Cornelis C (2010) Fuzzy-rough instance selection. In: Proceedings of the WCCI 2010 IEEE world congress on computational intelligence, IEEE congress on fuzzy logic, Barcelona, Spain, pp 1776–1782

Jensen R, Shen Q (2007) Fuzzy-rough sets assisted attribute selection. IEEE Trans Fuzzy Syst 15(1):73–89

Jensen R, Shen Q (2009) New approaches to fuzzy-rough feature selection. IEEE Trans Fuzzy Syst 17(4):824–838

Kim K (2006) Artificial neural networks with evolutionary instance selection for financial forecasting. Expert Syst Appl 30:519–526

Kira K, Rendell L (1992) A practical approach to feature selection. In: Proceedings of the 9th international workshop on machine learning, Aberdeen, Scotland UK, pp 249–256

Kohavi R, John G (1997) Wrappers for feature selection. Artif Intell 97:273–324

Kuncheva LI (1995) Editing for the *k*-nearest neighbors rule by a genetic algorithm. Pattern Recognit Lett 16:809–814

Kuncheva LI, Jain L (1999) Nearest neighbor classifier: simultaneous editing and descriptor selection. Pattern Recognit Lett 20:1149–1156

Kusunoki Y, Inuiguchi M (2010) A unified approach to reducts in dominance-based rough set approach. Soft Comput 14(5):507–515

Liu H, Motoda H (eds) (1998) Feature selection for knowledge discovery and data mining. The Springer international series in engineering and computer science, Springer, Berlin

Liu H, Motoda H (eds) (2001) Instance selection and construction for data mining. The Springer international series in engineering and computer science, Springer, Berlin

Liu H, Motoda H (eds) (2007) Computational methods of feature selection. Chapman & Hall/Crc data mining and knowledge discovery series, Chapman & Hall/Crc, London

Liu H, Yu L (2005) Toward integrating feature selection algorithms for classification and clustering. IEEE Trans Knowl Data Eng 17(3):1–12

Mjolsness E, DeCoste D (2001) Machine learning for science: state of the art and future prospects. Science 293:2051–2055

Oh IS, Lee JS, Moon BR (2004) Hybrid genetic algorithms for feature selection. IEEE Trans Pattern Anal Mach Intell 26:1424–1437

Pappa GL, Freitas AA (2009) Automating the design of data mining algorithms: an evolutionary computation approach. Natural computing. Springer, Berlin

Pawlak Z (1982) Rough sets. Int J Comput Inf Sci 11(5):341–356

Pawlak Z (1991) Rough sets: theoretical aspects of reasoning about data. Kluwer Academic Publishing, Dordrecht

Pawlak Z, Skowron A (2007a) Rough sets: some extensions. Inf Sci 177(1):28–40

Pawlak Z, Skowron A (2007b) Rudiments of rough sets. Inf Sci 177:3–27

Pyle D (1999) Data preparation for data mining. The Morgan Kaufmann series in data management systems. Morgan Kaufmann, Menlo Park

Quirino T, Kubat M, Bryan NJ (2010) Instinct-based mating in genetic algorithms applied to the tuning of 1-nn classifiers. IEEE Trans Knowl Data Eng 22(12):1724–1737

Radzikowska A, Kerre E (2002) A comparative study of fuzzy rough sets. Fuzzy Sets Syst 126:137–156

Ramentol E, Verbiest N, Bello R, Caballero Y, Cornelis C, Herrera F (2012) SMOTE-FRST: a new resampling method using fuzzy rough set theory. In: 10th International FLINS conference on uncertainty modelling in knowledge engineering and decision making (to appear)

Rokach L (2008) Genetic algorithm-based feature set partitioning for classification problems. Pattern Recognit 41:1676–1700

Saeys Y, Inza I, Larrañaga P (2007) A review of feature selection techniques in bioinformatics. Bioinformatics 19:2507–2517

Shakhnarovich G, Darrell T, Indyk P (eds) (2006) Nearest-neighbor methods in learning and vision: theory and practice. The MIT Press, Cambridge

Sheskin DJ (2011) Handbook of parametric and nonparametric statistical procedures, 5th edn. Chapman & Hall/CRC, London

Shie J, Chen S (2008) Feature subset selection based on fuzzy entropy measures for handling classification problems. Appl Intell 28: 69–82

Stracuzzi D, Utgoff P (2004) Randomized variable elimination. J Mach Learn Res 5:1331–1362

Triguero I, García S, Herrera F (2010) IPADE: Iterative prototype adjustment for nearest neighbor classification. IEEE Trans Neural Netw 21(12):1984–1990

Triguero I, Derrac J, García S, Herrera F (2012) A taxonomy and experimental study on prototype generation for nearest neighbor classification. IEEE Trans Syst Man Cybern Part C Appl Rev 42(1):86–100

Tsang E, Chen D, Yeung D, Wang X, Lee JT (2008) Attributes reduction using fuzzy rough sets. IEEE Trans Fuzzy Syst 16(5): 1130–1141

Weinberger K, Saul L (2009) Distance metric learning for large margin nearest neighbor classification. J Mach Learn Res 10: 207–244

Whitley LD (1989) The genitor algorithm and selection pressure: Why rank-based allocation of reproductive trials is best. In: Proceedings of the 3rd international conference on genetic algorithms, vol 2, Fairfax, Virginia, USA, June 1989, Morgan Kaufmann, pp 116–123

Wilson DL (1972) Asymptotic properties of nearest neighbor rules using edited data. IEEE Trans Syst Man Cybern 2(3):408–421

Wilson DR, Martinez TR (2000) Reduction techniques for instance-based learning algorithms. Mach Learn 38(3):257–286

Witten IH, Frank E, Hall MA (2011) Data mining: practical machine learning tools and techniques, 3rd edn. Morgan Kaufmann series in data management systems. Morgan Kaufmann, Menlo Park

Wu X, Kumar V (eds) (2009) The top ten algorithms in data mining. Data mining and knowledge discovery. Chapman & Hall/CRC, London

Zadeh LA (1965) Fuzzy sets. Inf Control 8(3):338–353

Zhai J (2011) Fuzzy decision tree based on fuzzy-rough technique. Soft Comput 15(6):1087–1096