Attribute and Case Selection for NN Classifier through Rough Sets and Naturally Inspired Algorithms

Yenny Villuendas-Rey[1] and Maria Matilde Garcia-Lorenzo[2]

[1] Department of Computer Science, University of Ciego de Avila, Cuba

[2] Department of Computer Science, University Marta Abreu of Las Villas, Cuba

yenny@informatica.unica.cu, mmgarcia@uclv.edu.cu

Abstract. Supervised classification is one of the most active research fields in the Artificial Intelligence community. Nearest Neighbor (NN) is one of the simplest and most consistently accurate approaches to supervised classification. The training set preprocessing is essential for obtaining high quality classification results. This paper introduces an attribute and case selection algorithm using a hybrid Rough Set Theory and naturally inspired approach to improve the NN performance. The proposed algorithm deals with mixed and incomplete, as well as imbalanced datasets. Its performance was tested over repository databases, showing high classification accuracy while keeping few cases and attributes.

Keywords. Nearest neighbor, case selection, attribute selection.

Selección de atributos y casos para el clasificador NN a través de conjuntos aproximados y algoritmos inspirados en la naturaleza

Resumen. La clasificación supervisada constituye una de las áreas de investigación más activas dentro de la Inteligencia Artificial. La regla del vecino más cercano (NN) es una de las más simples y efectivas para la clasificación supervisada. El pre-procesamiento del conjunto de entrenamiento es esencial para obtener clasificaciones de alta calidad. En este artículo se introduce un nuevo algoritmo de selección de atributos y casos que utiliza un enfoque híbrido basado en los Conjuntos Aproximados y los algoritmos inspirados en la naturaleza para mejorar el desempeño de clasificadores NN. El algoritmo propuesto permite el manejo de conjuntos de datos mezclados, incompletos, y no balanceados. El desempeño de dicho algoritmo se analizó utilizando bases de datos de repositorio, mostrando una alta eficacia del clasificador, utilizando solamente pocos casos y atributos.

Palabras clave. Vecino más cercano, selección de casos, selección de atributos.

1 Introduction

One of the simplest yet powerful techniques for case-based decision making is the Nearest Neighbor rule [1]. It is a non-parametric technique that uses the information of stored cases to determine the class of a new, unseen case. To do this, the Nearest Neighbor (NN) classifier compares the unknown instance to every case in the training set by means of a dissimilarity function. Then, it assigns it the class of its nearest case (nearest neighbor). Let $X = \{x_1, x_2, \cdots, x_n\}$ be a set of labeled cases, referred to as the training set of the NN classifier. Each case in X is described by a set $A = \{A_1, A_2, \cdots, A_m\}$ of attributes. If the attributes in A are of different type, for example, some are numerical and others are categorical, the cases in X have mixed descriptions. In addition, if an attribute value of some case is unknown, the description of the cases in X will be incomplete.

Dealing with mixed and incomplete cases in the context of NN has been approached in several ways. One of the most common is to codify categorical attributes and to fill missing values by means of some estimation procedures; and then to use some distance function, such as the Euclidean one, to compare the cases. One of the main disadvantages of this approach is that

codifying categorical attributes imposes an order relation among attribute values, which may not have sense in some domains. For example, attributes like color do not intrinsically have an order among attribute values. For instance, let $Color = \{red, green, yellow, pink\}$ be an attribute codified as $red = 1$, $green = 2$, $yellow = 3$, $pink = 4$, then the Euclidean distance between $(red, green) = 1$, and between $(red, pink) = 3$, which may have no sense in real life.

Another approach to deal with mixed and incomplete cases in the context of NN is to use dissimilarity functions able to compare cases with such characteristics. Usually, these dissimilarities use different comparison criterion for numeric and categorical attributes, and also for comparing cases with missing values. Among the most commonly used dissimilarities for mixed and incomplete data are HEOM and HVDM, introduced by Wilson and Martinez in 1997 [2]. Using this kind of dissimilarity function avoids modifying the original description of cases, but its main limitation is that this kind of dissimilarities does not fulfill the properties of a distance function (i.e., symmetric, positive defined and triangle inequality).

As known, the major disadvantages of the NN classifier are its storage and classification computational costs, which increase with the cardinality of the training set and the attribute set. To solve this drawback, the research community has followed two main alternatives: using distance properties to find the nearest neighbor of the unseen case without comparing it with the entire training set (known as fast nearest neighbor finding [3]), which is not directly applicable in mixed and incomplete domains given that dissimilarity functions do not fulfill distance properties; and preprocessing the training set by selecting relevant cases and attributes [4, 5]. This alternative has been found very useful to diminish NN costs and also to improve its accuracy.

Training set preprocessing by selecting both relevant attributes and cases is an active field of research. Studies carried out separately by Kuncheva and Jain [6] and Derrac *et al*. [7] have found that selecting relevant cases and attributes in a unique procedure leads to better results than sequential selection of attributes and cases.

This finding may be due to the fact that in sequential selection, the second algorithm does not access the (whole) original data, only the results of the first selection algorithm applied (Fig. 1).

Rough Set Theory (RST) has been widely used to separately select representative cases [8] and attributes [9]. RST is also a major component in some algorithms for integrated case and attribute selection. Section 2 presents the basics of RST. Other commonly used algorithms for attributes, cases, and integrated attribute and case selection are Swarm Intelligence and evolutionary algorithms (Section 3).

Although several proposals have been made to obtain a high quality preprocessed training set by integrated selection of cases and attributes, most of them are either not suitable for mixed and incomplete data having a strong stochastic component, or are intractable in high dimensionality domains. Section 4 introduces the IFIS-RST-SI algorithm for attribute and case selection, based on rough set theory and swarm intelligence. Section 5 covers some previous work done in selecting both cases and attributes for NN classifiers. The performance of the proposed algorithm is tested over repository datasets in Section 6, and conclusions and future works are given in Section 7.

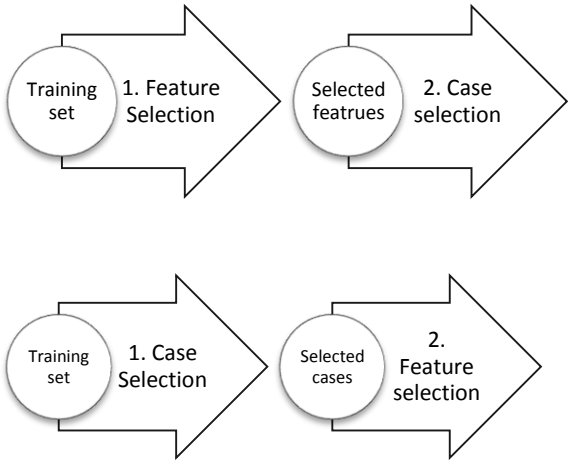


Fig. 1. In sequential selection of cases and attributes, the second selection algorithm only accesses the result of the first one

2 Rough Set Theory and Metaheuristic Algorithms in Training Set Selection for NN Classifiers

Rough Set Theory (RST) was introduced by Pawlak in 1982 [10]. RST tries to approximate any concept (a class, for example) by means of two sets: the lower and upper approximations of the concept. Let *X* be the set of labeled cases, described by the set of attributes *A*, the pair $(X, A \cup \{d\})$ where *d* is the decision attribute is called a Decision System.

In the classic RST, the lower and upper approximations of a class are constructed by using an indiscernibility relation over the attribute set. Two cases are discernible if they have different values on at least one attribute. The lower approximation of a class K_i is denoted by $INF(K_i)$ and is a set formed by all cases belonging to K_i which are discernible with respect to every other case not belonging to K_i, that is, $INF(K_i) = \{x \in K_i : [\forall y \notin K_i, \exists A_i \in A : x(i) \neq y(i)]\}$, where $x(i)$ and $y(i)$ are the values of the A_i attribute in cases *x* and *y*, respectively. On the other hand, in the upper approximation of a class K_i, denoted by $SUP(K_i)$ are included the cases belonging to the class, and also cases from different classes, but indiscernible with respect to cases of K_i, that is, $SUP(K_i) = \{x \in K_i \cup [y \notin K_i : \forall A_i \in A : x(i) = y(i)]\}$.The idea behind the lower and upper approximations is that cases in the lower approximation of a decision class are separate from cases belonging to other decision classes and therefore are sure members of the class, while cases in the upper approximation are possible members of the decision class.

For cases described by numerical attributes, constructing the lower approximation using the above definition may not be useful due to cases with very small differences between attribute values. Such cases are discernible, but they may not be separated enough from cases of different decision classes. To overcome this limitation, RST has been extended in order to use similarity (or dissimilarity) relations to decide if two cases are dissimilar enough to be considered as separated or discernible cases [11-13]. In the aforementioned works, the definitions of lower and upper approximations are rewritten in terms of dissimilarity relations. Thus, the lower approximation of a class will be $INF(K_i) = \{x \in K_i : [\forall y \notin K_i, diss(x, y) > \varepsilon]\}$, where $diss(x, y)$ is a dissimilarity function that compares cases *x* and *y*, and ε is a similarity threshold. In the same way, $SUP(K_i) = \{x \in K_i \cup [y \notin K_i : diss(x, y) \leq \varepsilon]\}$. The information of lower and upper approximations has been used to select representative cases of the training sets, and to eliminate possible noisy or mislabeled cases [8, 13, 14]. In addition, lower approximation has been also used for attribute selection, as an element of the Classification Quality measure [9, 11]. Classification quality (Equation 1) is computed as the amount of cases in the lower approximation of the decision with respect to the total amount of cases.

$$\gamma_A(X) = \frac{\left|\cup_{K_i \in K} INF(K_i)\right|}{|X|} \quad (1)$$

Let B⊆A be an attribute set, the higher the classification quality, the better the attribute set is. If $\gamma_B(X) = \gamma_A(X)$, then it is said that the attributes in B form a reduct of the decision system. In RST, attribute selection aims at obtaining an attribute set with the same discerning ability that the original attribute set, that is, a reduct. However, obtaining all possible reducts of a training set is an extremely costly computational process. Researches carried out by Bell and Guan [15] show that the computational cost of obtaining one reduct is bounded by $n^2 * m^2$, where *m* is the attribute set cardinality and *n* is the cardinality of the training set. They also found that the time complexity of finding all reducts is $O(2^m * J)$, where *J* is the cost of finding one reduct.

Although some efficient algorithms have been developed for finding the set of all reducts (for example, LEX [16]), they are unable to find them in training sets described by several tens of attributes. In addition, most RST based attribute selection algorithms obtain only one reduct [9, 17, 18].

The training set preprocessing by selecting cases or attributes can be viewed as a discrete optimization problem, where the search space is defined over all possible subsets of cases or attributes, and the optimization function is a measure of the selected training set quality.

Metaheuristic algorithms [19, 20] are suitable for this purpose.

Among metaheuristic algorithms, Genetic Algorithms (GAs) have been widely used for case selection [21-23] and for attribute selection [24-27]. Other metaheuristic algorithms used for case selection are Artificial Immune Systems [28, 29] and Particle Swarm Optimization [30]. They are also used for attribute selection [31-33]. Other metaheuristic algorithms used for this purpose are Ant Colony Optimization [34], Firefly Optimization [18] and Bee based optimization [9, 17].

Several fitness functions are proposed for case selection by means of metaheuristic algorithms, and most of them take into consideration classifier accuracy and cardinality of the selected case set. On the other hand, fitness functions for attribute selection usually use RST measures like classification quality (Equation 1). As shown, both metaheuristic algorithms and RST provide useful tools for training set preprocessing, for both attribute and case selection. In this paper, we use those approaches to select cases and attributes in large dimensionality domains.

3 Selecting Cases and Attributes for NN Classifiers by RST and Metaheuristic Algorithms

The proposed algorithm for case and attribute selection works in three stages (Fig. 2). Stage one preprocesses the training set by removing noisy, mislabeled or irrelevant cases. It also selects several relevant attributes sets. These processes are detailed in Sections 3.1 y 3.2, respectively. Stage two of the algorithm obtains several submatrices by projecting the selected cases using the attributes in the relevant attribute sets, and applying a case selection algorithm (Section 3.3). Finally, stage three merges the submatrices using a heuristic approach, also described in Section 3.3.

3.1 Training Case Preprocessing

Training set preprocessing is carried out according to the characteristics of the data. Among them, we take into consideration class

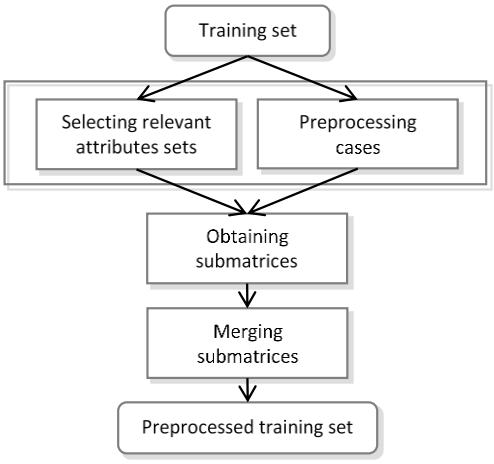
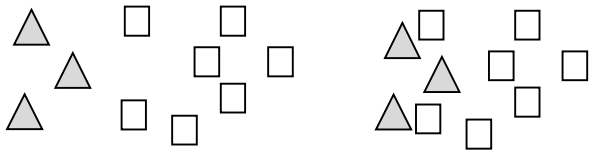


Fig. 2. Representation of proposed algorithm for case and attribute selection

overlapping and class imbalance. Removing noise or possible mislabeled cases is also an important step in training set preprocessing, given that it guarantees smoothing decision boundaries, and also may increase NN accuracy. We used Minimum Neighborhood Rough Sets to perform case filtering by selecting cases belonging to the lower approximation of the decision. Minimum Neighborhood Rough Sets [13] use similarity relations to decide if cases are distinguishable or not. Let $x \in X$ be a case of the training set and let $B \subseteq A$ be a set of attributes, the neighborhood $n_B(x)$ of x in the attribute space B is composed by cases having minimum dissimilarity with respect to x. Let $\mathrm{diss}^B(\mathrm{x},\mathrm{y})$ be a dissimilarity function, the neighborhood of a case x is given by $n_B(\mathrm{x}) = \left\{\mathrm{y} \in X : \mathrm{diss}^B(\mathrm{x},\mathrm{y}) = \min_{\mathrm{z} \in X}(\mathrm{diss}^B(\mathrm{x},\mathrm{z}))\right\}$. The lower approximation of a class K_i only includes cases whose neighborhood is composed by cases of its class, and also the cases that are not included in the neighborhoods of cases from different classes. It is defined as $INF(K_i) = \{x \in K_i : [\forall y \in n_B(x), y \in K_i]\}$.

Case filtering is carried out by selecting cases belonging to the lower approximation of the decision. However, some domains show a high degree of imbalance between decision classes. Class imbalance appears when one of the decision classes is much less represented than the others. NN classifiers should not be affected by class imbalance, but if decision classes are



(a) (b)

Fig. 3. Note how in (a) an NN using the Euclidean distance can achieve perfect classification of triangles, while in (b) no triangle can be correctly classified

highly overlapped, the minority class may be misclassified by an NN classifier (Fig. 3). In such domains, a filtering procedure may not obtain good results, due to the class overlap.

To deal with highly imbalanced and overlapped domains, our algorithm uses a preprocessing strategy consisting in preserving decision boundaries of classes, keeping cases belonging to the limit region, instead of cases belonging to the lower approximation. Figs. 4 and 5 offer the pseudocode of filtering and boundary preserving procedures for case preprocessing.

To select representative cases of a class having an empty limit region, we use a prototype selection procedure based on Maximum Similarity Graphs (MSG) [35]. An MSG is a directed graph that connects each case with its most similar neighbors. Formally, let $G = (X, \theta)$ be an MSG for a set of cases X, with arcs θ. In this graph, two cases $x, y \in X$ form an arc $(x, y) \in \theta$ if $\max_{z \in X}\{sim(x, z)\} = sim(x, y)$, where $sim(x, y)$ is a similarity function. Usually, $sim(x, y) = 1 - diss(x, y)$.

In case of ties, an MSG establishes a connection between a given case and each of its nearest neighbors. Compact Sets are the connected components of such graph.

The representative case selection procedure (Fig. 6) computes the Compact Sets of the classes having an empty limit region, and for each compact set, it selects as representative the one which maximizes the overall similarity between cases.

Intuition suggests filtering cases in balanced domains as well as in imbalanced, but not overlapped domains, and keeping border cases in imbalanced and highly overlapped scenarios. However, we use the JRip classifier available with the Weka software [36] to obtain a set of rules to decide whether to filter or to preserve boundaries in case preprocessing.

The rule obtained by JRip was the following: if IR > 10 and CO > 0.75, preserve boundaries, else

Case filtering procedure (CF)

Input: Training set X, dissimilarity function diss

1. P = ϕ
2. Compute the lower approximation of the decision as $\mathrm{INF(X)} = \cup_{\mathrm{i}} \mathrm{INF(K_i)}$, where $INF(K_i) = \{x \in K_i : [\forall y \in n_B(x), y \in K_i]\}$
3. Keep the cases included in the lower approximation, $P = \mathrm{INF(X)}$

Return P

Fig. 4. Procedure for filtering cases in training case preprocessing

Boundary preserving procedure (CB)

Input: Training set X, dissimilarity function diss

1. P = ϕ
2. For each decision class K_i
   2.1. Compute the limit region of the decision in a Nearest Neighborhood Rough Set, as $LIM(K_i) = SUP(K_i) - INF(K_i)$.
   2.2. If the limit region of the class is empty, then $P = P \cup Representative\_Cases(\mathrm{K_i})$. Else, $P = \mathrm{P} \cup LIM(K_i)$

Return P

Fig. 5. Procedure for boundary preserving in training case preprocessing

Representative case procedure

Input: decision class K_i, dissimilarity function diss

1. $R = \phi$
2. Compute compact sets (CS) of K_i
   2.1. For each $cs \in CS$
      2.1.1. Select a representative case *r* as
      $$\mathrm{r} = \underset{o \in CS}{\operatorname{argmin}} \left\{ \sum_{i \in CS} diss(o, i) \right\}$$
      2.1.2. $R = R \cup \{r\}$

Return *R*

Fig. 6. Procedure for selecting representative cases in decision classes having an empty limit region

0	1	1	0	0	1

Fig. 7. Representation of a candidate attribute set (food source) in the ABC algorithm. Only the second, third and sixth attributes are included in the selected attribute set.

filter cases. Imbalance ratio (IR) is computed as the case count belonging to minority class, with respect to cases belonging to the majority class. Class overlapping (CO) is measured as the maximum amount of cases of a class having nearest neighbors of different class, that is, cases not belonging to the lower approximation of Minimum Neighborhood Rough Sets. Although the thresholds for IR and CO are fixed by the JRip rule, this rule can be interpreted as follows: if IR is high and CO is high, preserve boundaries, otherwise filter cases.

3.2 Selecting Relevant Attribute Sets

Using a unique attribute set in case and attribute selection may not consider some interactions among attributes, and may exclude some relevant features, also computing all reducts is impracticable in domains with several tens of features. To overcome this drawback, we propose to use population–based metaheuristic algorithms to obtain several candidate attribute sets. We explore using Artificial Bee Colonies and Genetic Algorithms for this purpose.

Artificial Bee Colony (ABC) optimization was introduced by Karaboga and Basturk [37-39]. This metaheuristic procedure is based on the behavior of honey bees for finding good flower nectar. In a bee colony, there are three kinds of bees: scout bees, employed bees and onlooker bees. Scout bees search the area around the colony to find food sources for all the other bees. Once found, employed bees are assigned to food sources to extract the nectar.

Each employed bee takes a portion of nectar and return to the colony. Then, onlooker bees taste the nectar portions provided by employed bees, and select the best food sources based on the nectar quality. Then, onlooker bees search around the best food sources, to obtain other sources with higher quality. This process

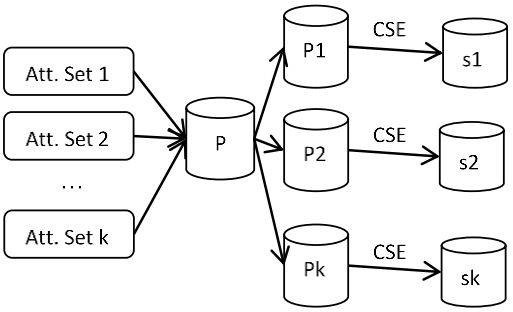


Fig. 8. Representation of the process for obtaining submatrices

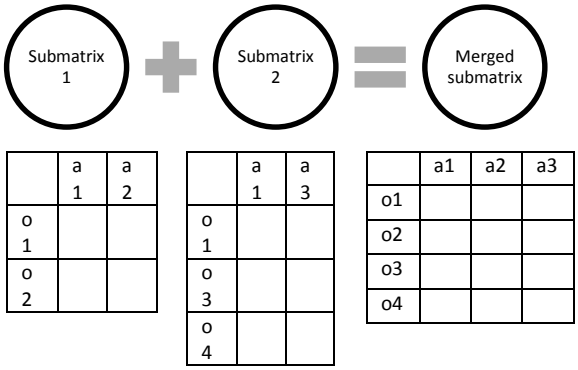


Fig. 9. Result of merging two submatrices

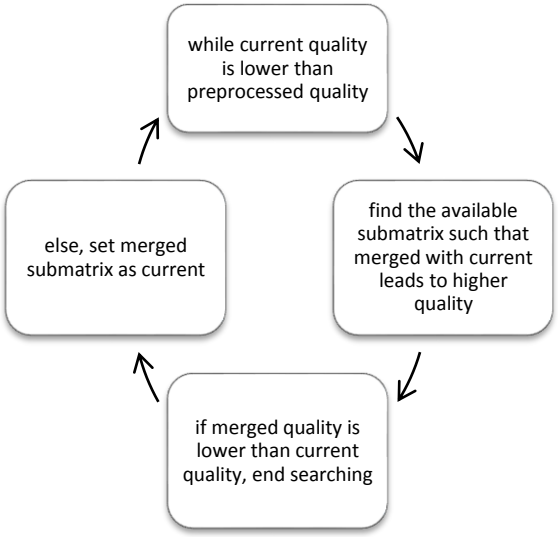


Fig. 10. Submatrix merging strategy

Table 1. Datasets used in experiments

Dataset	Attributes	Cases	IR	CO
breast-cancer*	9	289	2.37	0.74
breast-w	9	699	1.90	0.10
car	6	1728	18.69	1.00
colic*	22	368	1.73	0.29
credit-a*	15	690	1.25	0.23
credit-g	20	1000	2.35	0.53
cylinder *	40	512	1.37	0.36
dermatology*	34	366	5.62	0.21
diabetes	8	768	1.87	0.46
heart-c*	13	303	1.20	0.28
heart-h*	13	294	1.77	0.32
hepatitis*	19	155	3.87	0.51
hypothyroid*	30	3772	1497.35	1.00
kr-vs-kp	36	3196	1.09	0.32
lymph	18	148	47.55	0.64
mushroom*	22	8124	1.08	0.00
page-blocks	10	5473	175.99	0.48
sick*	30	3772	15.12	0.41
spambase	57	4601	1.54	0.12
splice	60	3190	2.17	0.45
tic-tac-toe	9	958	1.89	0.94
vehicle	18	946	1.10	0.53
vowel	12	990	1.12	0.03
waveform	21	5000	1.03	0.30
wine*	13	178	1.47	0.13

Datasets marked with * have missing values.

guarantees that the colony exploits the best food sources and obtains high quality nectar. Once a food source is exhausted, scout bees replace it by a new one.

As the ABC algorithm maintains a population of food sources, it is suitable for obtaining several attribute sets by returning the entire population. For attribute selection, food sources are modeled as binary strings having as many bits as attributes. Each bit represents the inclusion or exclusion of a certain attribute in the solution (Fig. 7).

In the ABC algorithm, each food source is associated to a nectar quality, which guides the searching procedure. We use the RST measure for classification quality (Equation 1) as the amount of nectar of candidate attribute sets, but computing lower approximation by using Minimum Neighborhood Rough Sets [13].

To obtain a food source near to a current source, we use a modification strategy consisting in changing bits in the string representing the current source.

Genetic algorithms maintain a population of candidate solutions and are suitable for obtaining multiple attribute sets. We also explore using GA based attribute selection algorithms (GMAS) to obtain multiple attribute sets.

GMAS uses the same codification strategy of BMAS (binary strings), and also uses the classification quality measure (Equation 1) as the fitness function. GMAS uses an elitist selection strategy. In crossover, GMAS selects parents through tournament, and it uses single point crossover to obtain the children.

3.3 Obtaining and Merging Submatrices

The first stage of the algorithm for selecting both attributes and cases has as outcome a preprocessed case set P, and a list of relevant attribute sets, MAS.

In stage two, the cases belonging to the set *P* are projected using the attribute set corresponding to each item of the MAS list, having as many projections P_i as attribute sets in the MAS list. Once the projections are obtained, a subclass consistent case selection procedure is applied to each projection, and several submatrices s_i are obtained (Fig. 8). We suggest using the CSE algorithm [40] for case selection in the projections. CSE is a compact set based algorithm for case selection, able to deal with mixed and incomplete data; it is subclass consistent, and also preserves the inner structure of decision classes. The third stage of the algorithm consists in integrating the information of the submatrices through a merging procedure. Fig. 9 illustrates the merging of two submatrices. Note how the merged submatrix contains all cases and attributes of the original ones. The algorithm uses a greedy approach for merging (Fig. 10). First, it sorts the submatrices according to classification quality (Equation 1). Then, if the first (best) submatrix has better quality than the preprocessed training set, the process ends.

If not, the procedure will find the available submatrix that merged to the current one, which leads to a better quality. If the merging is not successful or there is no available submatrix, the process ends. Otherwise, the merging cycle is repeated until the desired quality is achieved or one of the stopping conditions is reached.

The algorithm handles mixed and incomplete data, and allows selecting several candidate attribute sets in large dimensionality domains, due to the use of metaheuristic procedures. It employs training set preprocessing procedures according to the characteristics of data. The algorithm also uses an RST measure, the classification quality, to guide the attribute set selection process and the submatrix sorting and merging procedures.

4 Previous Works on Attribute and Case Selection for NN Classifiers

The selection of cases and attributes in a unique procedure was first proposed by Skalak [41] who used Random Mutation Hill Climbing search strategy for this purpose, and introduced the RMHC-FP1 algorithm. A few years later, Kuncheva and Jain [6], as well as Ishibushi and Nakashima [42] used genetic algorithms as the underlying search strategy for selecting both attributes and cases. Another GA based algorithms for this purpose are the ones by Ahn *et al*. [43, 44] and by Rozypal and Kubat [45]. Recently, evolutionary algorithms are hybridized with other techniques to improve classifier accuracy and to obtain a reduced training set. Examples of hybrid algorithms are HG [46], EIS-RFS [7] and EFS-RPS [47].

The first deterministic algorithm for selecting attributes and cases is the one proposed by Dasarathy in 2000 [48]. It combines the Sequential Backward Search (SBS) [49] algorithm for attribute selection with the application of two case selection algorithms: the Proximity Graph Based Editing using Relative Neighborhood graphs (RNG-E) [50] and the Minimal Consistent Subset method (MCS) [51]. The proposal by Dasarathy uses as the fitness function for the SBS algorithm a combined measure of the 1-NN accuracy with respect to a validation set and the amount of case reduction achieved by the sequential application of the case selection methods. To combine the accuracy and reduction, the method uses the Euclidean distance. This algorithm is computationally expensive, and also may delete an entire class in a training set.

In 2006, Villuendas *et al*. introduced another deterministic algorithm, SOFSA [52], specially

designed to handle mixed and incomplete data. It is based on reduct computation and uses the CSE algorithm for case selection [40]. The IFIS algorithm [53] is inspired in SOFSA and uses RST to carry out an intelligent preprocessing of the training set. However, computing all reducts is impracticable on high dimensionality training sets, making IFIS and SOFSA not adequate to deal with this kind of data.

5 Numerical Experiments

This section presents the experiments carried out to test the performance of the proposed algorithms. It covers the data sets and dissimilarity function used in our research, as well as the algorithms used in the comparisons and the statistical procedures applied to contrast the results. In addition, this section details the results and their analysis.

5.1 Data Sets and Algorithms

We used 25 datasets from the Machine Learning Repository of University of California at Irvine [54]. Their description is given in Table 1, considering the amount of attributes (Attributes), the amount of cases (Cases), the imbalance ratio (IR) and the class overlapping (CO).

We use the HEOM dissimilarity proposed by Wilson and Martinez [2] (Equation 2), which is able to handle mixed attribute types as well as missing values.

$$HEOM(x,y) = \sqrt{\sum_{a=1}^{m} d_a(x_a, y_a)}$$

$$d_a = \begin{cases} 1 \\ overlap(x_a, y_a) \\ diff(x_a, y_a) \end{cases} \quad (2)$$

$$overlap(x,y) = \begin{cases} 0 \text{ if } x = y \\ 1 \text{ otherwise} \end{cases}$$

$$diff(x,y) = |x - y| / max_a - min_a$$

The datasets are partitioned into subsets by using the 10 fold cross-validation procedure. Each dataset is partitioned into 10 equally sized subsets, preserving class distribution. In a repetitive process, nine subsets are used for training, and the remaining is used for testing. The final results of the algorithms are computed by averaging the results over the ten partitions. In addition, stochastic algorithms are applied three times, and the results are averaged.

We carried out three experiments. First, we compare the performance of the proposed BMAS and GMAS algorithms for multiple attribute selection, with respect to computation of all reducts, using the LEX algorithm [16]. Then, we analyze the impact of the proposed algorithm for selecting attributes and cases in the performance of NN classifier, and we compare it with selecting only attributes or only cases.

In the last experiment, we compare the proposed algorithms with respect to other proposals. We selected the algorithms proposed by Dasarathy (DS) [48], Derrac *et al*. (EIS-RFS) [7] and Villuendas *et al*. (IFIS) [53], considered as representative of the state of the art in selecting attributes and cases. Table 2 details the parameters used in the different algorithms.

All experiments were carried out using a laptop with the Windows Seven operating system, running on an AMD Sempron SI-42 microprocessor at 2.10 GHz, and with 2.75 GB of usable RAM. If an algorithm spends more than 24 hours without obtaining a result, we abort its execution. These situations are marked in tables with the label "+24".

We use hypothesis testing to establish the existence or non-existence of significant differences in the performance of the compared algorithms. We used Wilcoxon's test recommended by Demsar [55].

Table 2. Parameters used by the algorithms

Algorithms	Parameters
BMAS	Food sources: 10, Evaluations: 1000, Onlooker Bees: 3, Tournament size: 30%, Limit: 20
GMAS	Population: 10, Evaluations: 1000, Crossover probability: 1.0, Mutation probability: 0.05 per bit, Tournament size: 30%
EIS-RFS	Population: 10, Evaluations: 1000, Crossover probability: 1.0, Mutation probability: 0.05 per bit, alpha: 0.5, MaxGamma: 1.0, UpdateFS: 10, beta: 0.75

Table 3. Averaged AUC obtained by BMAS, GMAS and LEX

Datasets	LEX	GMAS	BMAS
breast-cancer	0.548	0.557	**0.565**
breast-w	**0.943**	0.939	0.937
car	0.594	**0.629**	0.525
colic	**0.783**	0.774	0.770
credit-a	**0.819**	0.817	0.791
credit-g	**0.621**	0.611	0.596
cylinder	+24	0.736	**0.740**
dermatology	+24	0.912	**0.937**
diabetes	**0.675**	0.658	0.638
heart-c	**0.773**	0.770	0.766
heart-h	0.727	**0.755**	0.745
hepatitis	0.637	**0.669**	0.618
hypothyroid	0.552	**0.564**	0.547
kr-vs-kp	0.924	**0.943**	0.771
lymph	0.798	**0.922**	0.874
mushroom	**1.000**	**1.000**	**1.000**
page-blocks	0.773	**0.779**	0.142
sick	0.786	**0.828**	0.672
spambase	+24	**0.880**	**0.880**
splice	+24	**0.768**	0.751
tic-tac-toe	0.746	0.707	**0.747**
vehicle	0.686	0.703	**0.712**
vowel	**0.995**	0.978	0.991
waveform	+24	**0.700**	0.688
wine	**0.962**	0.951	0.954
Times Best	**9**	**12**	**7**

We compute case retention (CR) and attribute retention (AR) as the quality measures of performance. Given that we are dealing with some imbalanced datasets, we used the Area under ROC curve (AUC) to evaluate classifier performance on testing sets. AUC is a quality measure widely used to evaluate classifiers in imbalanced domains [56]. To compute the AUC for a discrete classifier, a simple method is proposed in [57], based on a confusion matrix (Equation 3).

$$AUC(x,y) = \frac{1}{|K|}\sum_{K_i \in K} \frac{WellClassifed(K_i)}{|K_i|} \qquad (3)$$

Table 4a. Results of Wilcoxon's test comparing attribute set selection algorithms according to AUC

Pair	w-l-t	probability
GMAS vs. LEX	10-9-1	0.235
BMAS vs. LEX	5-14-1	0.049
GMAS vs. BMAS	16-7-2	0.019

Table 4b. Results of Wilcoxon's test comparing attribute set selection algorithms according to AR

Pair	w-l-t	probability
GMAS vs. LEX	17-3-0	0.004
BMAS vs. LEX	18-2-0	0.002
GMAS vs. BMAS	9-16-0	0.187

where $WellClassifed(K_i)$ returns the count of cases belonging to class K_i that are correctly classified.

5.2 BMAS vs. GMAS for Selecting Attributes

Both genetic algorithms and artificial bee colony optimization are population–based metaheuristic procedures. However, they have different search strategies and they do exploration and exploitation of solutions in a different manner. In this section, we compare the performance of the BMAS and GMAS algorithms for attribute sets selection with respect to computation of all reducts using the LEX algorithm.

Table 4a shows the AUC results achieved by the algorithms. For each algorithm, we compute the average AUC of all obtained attribute sets, and also the 10 fold cross validation average (best results in each datasets are highlighted in bold). We refer to computation of all reducts as LEX. Table 4b shows the results according to attribute retention (AR). The tables also show the number of times each algorithm achieves the best results.

The results show the validity of the proposed GMAS algorithm for multiple attribute sets selection, as it obtains the best results according to AUC. The LEX algorithm was unable to obtain the reducts in some databases, spending more than 24 hours without obtaining any result.

Table 5. Averaged attribute retention (AR) obtained by BMAS, GMAS, and LEX

Datasets	LEX	GMAS	BMAS
breast-cancer	0.867	0.943	**0.851**
breast-w	1.000	0.712	**0.704**
car	1.000	**0.733**	0.827
colic	0.738	**0.555**	0.585
credit-a	0.873	0.665	**0.601**
credit-g	0.634	**0.588**	0.604
cylinder	+24	0.531	**0.523**
dermatology	+24	0.659	**0.651**
diabetes	1.000	0.650	**0.640**
heart-c	0.831	0.648	**0.625**
heart-h	0.908	0.655	**0.637**
hepatitis	0.674	0.575	**0.547**
hypothyroid	0.862	0.571	**0.492**
kr-vs-kp	0.772	0.668	**0.549**
lymph	**0.475**	0.724	0.692
mushroom	**0.300**	0.576	0.536
page-blocks	0.990	0.637	**0.116**
sick	0.759	**0.475**	0.526
spambase	+24	**0.525**	0.526
splice	+24	**0.516**	0.522
tic-tac-toe	0.889	**0.659**	0.682
vehicle	0.933	0.591	**0.588**
vowel	0.846	**0.632**	0.780
waveform	+24	**0.483**	0.512
wine	0.717	0.614	**0.583**
Times Best	**2**	**9**	**14**

However, the Wilcoxon's test confirms that differences between GMAS and LEX in AUC are not significant at the 0.01 significance level.

According to attribute retention, the BMAS algorithm obtains the best results. It outperforms other algorithms on most databases. Also, GMAS keeps less attributes than LEX. Moreover, the Wilcoxon's test confirms that differences between BMAS and LEX (and also between GMAS and LEX) are significant at the 0.01 significance level. The test found no significant differences between GMAS and BMAS.

5.3 IFIS-RST-GA in NN Preprocessing

As the proposed GMAS algorithm obtains best results according to AUC, and also good results in attribute retention, we used it for selecting multiple attribute sets in the proposed algorithm for both case and attribute selection. In this section, we compare the performance of the proposed IFIS-RST-GA algorithm with respect to attribute selection (using GMAS), case selection

Table 6. Averaged AUC obtained by NN, GMAS, CP and IFIS-GA

Datasets	NN	GMAS	CP	IFIS-GA
breast-cancer	0.565	0.557	0.569	**0.586**
breast-w	0.943	0.939	**0.965**	0.950
car	0.594	**0.629**	0.592	0.250
colic	0.781	0.774	0.768	**0.784**
credit-a	0.812	0.817	0.849	**0.857**
credit-g	**0.630**	0.611	0.618	0.597
cylinder	**0.751**	0.736	0.673	0.687
dermatology	**0.940**	0.912	0.937	0.922
diabetes	0.675	0.658	**0.700**	0.699
heart-c	0.774	0.770	**0.815**	0.784
heart-h	0.745	0.755	0.777	**0.808**
hepatitis	0.632	0.669	**0.681**	0.608
hypothyroid	0.552	**0.564**	0.495	0.484
kr-vs-kp	0.901	**0.943**	0.849	0.923
lymph	0.888	**0.922**	0.663	0.711
mushroom	**1.000**	**1.000**	**1.000**	0.500
page-blocks	0.773	**0.779**	0.728	0.735
sick	0.783	**0.828**	0.732	0.761
spambase	**0.905**	0.880	0.888	0.877
splice	**0.797**	0.768	0.772	0.750
tic-tac-toe	0.764	0.707	**0.821**	0.709
vehicle	**0.687**	0.703	0.682	0.682
vowel	**0.995**	0.978	0.974	0.932
waveform	0.733	0.700	**0.759**	0.699
wine	**0.962**	0.951	0.960	0.947
Times Best	**9**	**7**	**7**	**4**

Table 7. Results of Wilcoxon's test comparing IFIS-GA vs. NN, GMAS and CP algorithms according to AUC

Pair	w-l-t	probability
IFIS-GA vs. NN	8-17-0	0.012
IFIS-GA vs. GMAS	9-16-0	0.049
IFIS-GA vs. CP	9-15-1	0.166

Table 8. Averaged attribute reduction (AR) and case reduction (CR) obtained by GMAS, CP and IFIS-GA

Datasets	AR		CR	
	GMAS	IFIS-GA	CP	IFIS-GA
breast-cancer	**0.943**	0.944	0.491	**0.207**
breast-w	0.712	**0.711**	0.944	**0.217**
car	0.733	**0.700**	0.750	**0.040**
colic	**0.555**	**0.555**	0.784	**0.297**
credit-a	0.665	**0.660**	0.813	**0.353**
credit-g	0.588	**0.570**	0.708	**0.346**
cylinder	**0.531**	0.538	0.775	**0.373**
dermatology	**0.659**	**0.659**	0.942	**0.384**
diabetes	0.650	**0.625**	0.705	**0.338**
heart-c	0.648	**0.646**	0.760	**0.337**
heart-h	0.655	**0.654**	0.778	**0.344**
hepatitis	**0.575**	0.584	0.807	**0.314**
hypothyroid	**0.571**	0.579	**0.408**	**0.408**
kr-vs-kp	**0.668**	0.675	0.685	**0.224**
lymph	**0.724**	0.728	0.777	**0.342**
mushroom	0.576	**0.277**	1.000	**0.032**
page-blocks	0.637	**0.630**	0.956	**0.403**
sick	**0.475**	0.479	0.959	**0.375**
spambase	**0.525**	0.691	0.908	**0.630**
splice	**0.516**	0.518	0.688	**0.347**
tic-tac-toe	**0.659**	0.700	0.164	**0.061**
vehicle	0.591	**0.539**	0.696	**0.334**
vowel	0.632	**0.631**	0.879	**0.434**
waveform	0.483	**0.440**	0.990	**0.376**
wine	**0.614**	0.623	0.735	**0.388**
Times Best	**13**	**14**	**1**	**25**

Table 9. Results of Wilcoxon's test comparing IFIS-GA vs. GMAS and CP algorithms according to AR and CR

Pair	w-l-t	probability
IFIS-GA vs. GMAS	12-11-2	0.761
IFIS-GA vs. CP	24-0-1	0.000

(using case preprocessing, CP, procedures of IFIS-GA) and with respect to NN classifier using all cases and attributes.

Tables 6 and 8 show the results achieved by the algorithms according to AUC and attributes and cases reduction, respectively. In addition,

Table 10. Averaged AUC obtained by DS, EIS-RFS, IFIS and IFIS-GA

Datasets	DS	EIS-RFS	IFIS	IFIS-GA
breast-cancer	0.572	0.538	**0.604**	0.586
breast-w	0.947	0.950	0.956	0.950
car	0.331	0.344	**0.763**	0.250
colic	0.759	0.696	**0.790**	0.784
credit-a	0.817	0.776	0.841	**0.857**
credit-g	0.598	0.522	**0.599**	0.597
cylinder	0.656	0.632	+24	**0.687**
dermatology	0.896	0.901	+24	**0.922**
diabetes	0.701	0.654	**0.708**	0.699
heart-c	0.769	0.767	0.020	**0.784**
heart-h	0.788	0.772	0.786	**0.808**
hepatitis	0.696	0.661	**0.719**	0.608
hypothyroid	+24	+24	0.469	**0.484**
kr-vs-kp	+24	+24	0.902	**0.923**
lymph	0.703	0.601	0.675	**0.711**
mushroom	+24	+24	**1.000**	0.500
page-blocks	+24	+24	**0.746**	0.735
sick	+24	+24	0.712	**0.761**
spambase	+24	+24	+24	**0.877**
splice	+24	+24	+24	**0.750**
tic-tac-toe	0.664	0.532	0.666	**0.709**
vehicle	0.670	0.491	**0.694**	0.682
vowel	**0.946**	0.198	0.889	0.932
waveform	+24	+24	+24	0.699
wine	0.912	0.848	0.935	**0.947**
Times Best	**1**	**0**	**9**	**13**

Table 11. Results of Wilcoxon's test comparing IFIS-GA vs. DS, EIS-RFS and IFIS according to AUC

Pair	w-l-t	probability
IFIS-GA vs. DS	12-5-0	0.107
IFIS-GA vs. EIS-RFS	14-2-1	0.008
IFIS-GA vs. IFIS	10-10-0	0.467

Tables 7 and 9 detail the statistical comparison among the algorithms using Wilcoxon's test.

The proposed IFIS-GA obtains slightly worse performance than the original NN according to AUC and GMAS.

However, the Wilcoxon's test confirms that differences between IFIS-GA and other

Table 12. Averaged attribute retention (AR) obtained by DS, EIS-RFS, IFIS and IFIS-GA

Datasets	DS	EIS-RFS	IFIS	IFIS-GA
breast-cancer	**0.933**	1.000	0.867	0.944
breast-w	0.922	1.000	1.000	**0.711**
car	1.000	1.000	1.000	**0.700**
colic	0.955	1.000	0.732	**0.555**
credit-a	0.940	1.000	0.873	**0.660**
credit-g	0.950	1.000	0.625	**0.570**
cylinder	0.974	0.988	+24	**0.538**
dermatology	0.971	0.879	+24	**0.659**
diabetes	0.630	1.000	1.000	**0.625**
heart-c	0.939	1.000	1.000	**0.646**
heart-h	0.939	1.000	0.908	**0.654**
hepatitis	0.947	1.000	0.658	**0.584**
hypothyroid	+24	+24	0.862	**0.579**
kr-vs-kp	+24	+24	0.772	**0.675**
lymph	0.956	1.000	**0.483**	0.728
mushroom	+24	+24	0.568	**0.277**
page-blocks	+24	+24	0.990	**0.630**
sick	+24	+24	0.759	**0.479**
spambase	+24	+24	+24	**0.691**
splice	+24	+24	+24	**0.518**
tic-tac-toe	0.889	1.000	0.889	**0.700**
vehicle	0.944	0.856	0.933	**0.539**
vowel	0.931	1.000	0.846	**0.631**
waveform	+24	+24	+24	**0.440**
wine	0.923	1.000	0.731	**0.623**
Times Best	**1**	**0**	**1**	**23**

Table 13. Results of Wilcoxon's test comparing IFIS-GA vs. DS, EIS-RFS and IFIS algorithms according to attribute retention

Pair	w-l-t	probability
IFIS-GA vs. DS	16-1-0	0.000
IFIS-GA vs. EIS-RFS	17-0-0	0.000
IFIS-GA vs. IFIS	18-2-0	0.001

algorithms in AUC are not significant at the 0.01 significance level.

According to attribute retention and case retention, the proposed IFIS-GA obtains the best results. The Wilcoxon's test confirms that differences between IFIS-GA and case preprocessing algorithms are significant at the 0.01 significance level. However, the differences between IFIS-GA and GMAS are not significant.

Although IFIS-GA obtains slightly worse performance than the original NN according to AUC, the results show that using both attributes and cases in a unique procedure gives better results than selecting only cases or only attributes.

5.4 IFIS-RST-GA vs. Other Algorithms

In this section, we compared the performance of IFIS-GA with previously reported algorithms for selecting both cases and attributes. We selected the DS [48], EIS-RFS [7], and IFIS [53] algorithms, because they are representative of different approaches for hybrid selection.

Tables 10, 12, and 14 show the averaged results of the algorithms according to AUC, attribute retention and case retention, respectively. The statistical comparisons of these experimental results are presented in Tables 11, 13, and 15.

The results show the validity of the proposed IFIS-GA algorithm for selecting cases and attributes, as it obtains the best results according to AUC.

The IFIS algorithm was unable to obtain any result because the reduct computation in some databases takes more than 24 hours. However, the Wilcoxon's test confirms that differences between IFIS-GA and IFIS in AUC are not significant at the 0.05 significance level (Table 11).

The proposed algorithm also obtains the best results according to attribute retention (Table 12), keeping much less attributes than other algorithms.

The Wilcoxon's test (Table 13) confirms that differences between IFIS-GA and all other algorithms in attribute retention are significant at the 0.01 significance level.

According to cases retention, the EIS-RFS algorithm obtains the best results, keeping less than 10% of cases in each dataset. The proposed IFIS-GA outperforms IFIS according to cases retention, but loses with respect to DS and EIS-RFS.

Table 14. Averaged case retention (CR) obtained by DS, EIS-RFS, IFIS and IFIS-GA

Datasets	DS	EIS-RFS	IFIS	IFIS-GA
breast-cancer	0.157	**0.021**	0.220	0.207
breast-w	0.030	**0.015**	0.267	0.217
car	0.020	**0.012**	0.343	0.040
colic	0.107	**0.020**	0.293	0.297
credit-a	0.101	**0.013**	0.367	0.353
credit-g	0.155	**0.010**	0.362	0.346
cylinder	0.187	**0.030**	+24	0.373
dermatology	0.124	**0.086**	+24	0.384
diabetes	0.146	**0.013**	0.308	0.338
heart-c	0.140	**0.014**	0.183	0.337
heart-h	0.095	**0.015**	0.344	0.344
hepatitis	0.067	**0.033**	0.358	0.314
hypothyroid	+24	+24	**0.378**	0.408
kr-vs-kp	+24	+24	0.246	**0.224**
lymph	0.238	**0.034**	0.381	0.342
mushroom	+24	+24	0.116	**0.032**
page-blocks	+24	+24	**0.397**	0.403
sick	+24	+24	0.393	**0.375**
spambase	+24	+24	+24	**0.630**
splice	+24	+24	+24	**0.347**
tic-tac-toe	0.298	**0.011**	0.071	0.061
vehicle	0.200	**0.015**	0.309	0.334
vowel	0.197	**0.010**	0.179	0.434
waveform	+24	+24	+24	**0.376**
wine	0.117	**0.035**	0.380	0.388
Times Best	**0**	**17**	**2**	**6**

Table 15. Results of Wilcoxon's test comparing IFIS-GA vs. DS, EIS-RFS and IFIS algorithms according to case retention

Pair	w-l-t	probability
IFIS-GA vs. DS	1-15-0	0.003
IFIS-GA vs. EIS-RFS	0-17-0	0.000
IFIS-GA vs. IFIS	11-8-1	0.398

The results show good performance of the proposed IFIS-GA. It obtains better or equal results than previously reported algorithms. It also has a computational complexity lower than the other methods.

6 Conclusions

Preprocessing of a training set is very important for NN classifiers allowing them to obtain high quality classifiers by removing noisy or mislabeled cases and redundant or irrelevant attributes.

This paper introduced a new attribute and case selection algorithm called IFIS-GA. The proposal is based on hybridizing Rough Set Theory and naturally inspired algorithms to obtain a reduced training set, in both attributes and cases. The performance of the IFIS-GA algorithm was tested over twenty five repository datasets. The experimental results show that the proposed Genetic Multiple Feature Selection algorithm is useful to obtain several attribute sets, leading the proposed IFIS-GA to obtain better results than previous algorithms.

The experiments also show that using the classification quality measure of RST in the sorting and merging strategies of IFIS-GA leads to better results according to area under the ROC curve, compared with other algorithms.

References

1. **Cover, T. & Hart, P. (1967).** Nearest neighbor pattern classification. *IEEE Transactions on Information Theory.* 13(1), 21–27.
2. **Wilson, D.R. & Martinez, T.R. (1997).** Improved Heterogeneous Distance Functions. *Journal of Artificial Intelligence Research*, 6(1), 1–34.
3. **Zhang, B. & Shrihari, S.N. (2004).** Fast k-nearest neighbors Classification using cluster-based trees. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(4), 525–528.
4. **Liu, H., Motoda, H., Setiono, R., & Zhao, Z. (2010).** Feature selection: an ever evolving frontier in data mining. *JMLR Workshop and Conference Proceedings*, Hyderabad, India, 10, pp. 4–13.
5. **García, S., Derrac, J., Cano, J., & Herrera, F. (2012).** Prototype selection for nearest neighbor classification: Taxonomy and empirical study. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(3), 417–435.
6. **Kuncheva, L.I. & Jain, L.C. (1999).** Nearest neighbor classifier: Simultaneous editing and feature selection. *Pattern Recognition Letters*, 20(11-13), 1149–1156.

7. **Derrac, J., Cornelis, C., García, S., & Herrera, F. (2012).** Enhancing evolutionary instance selection algorithms by means of fuzzy rough set based feature selection. *Information Sciences*, 186(1), 73–92.
8. **Caballero, Y., Bello, R., Salgado, Y., & García, M.M. (2007).** A method to edit training sets based on rough sets. *International Journal of Computational Intelligence Research*, 3(3), 219–229.
9. **Fan, H. & Zhong, Y. (2012).** A rough set approach to feature selection based on wasp swarm optimization. *Journal of Computational Information Systems,* 8(3), 1037–1045.
10. **Pawlak, Z. (1982).** Rough sets. *International Journal of computer and Information Sciences*, 11(5), 341–356.
11. **Hu, Q., Yu, D., Liu, J., & Wu, C. (2008).** Neighborhood rough set based heterogeneous feature selection. *Information Sciences*, 178(18), 3577–3594.
12. **Filiberto, Y., Caballero, Y., Larrua, R., & Bello, R. (2010).** A method to build similarity relations into extended rough set theory. *10th International conference on Intelligent Systems Design and Applications (ISDA 2010),* Cairo, Egypt, 1314–1319.
13. **Villuendas-Rey, Y., Caballero-Mota, Y., & García-Lorenzo, M.M. (2012).** Using Rough Sets and Maximum Similarity Graphs for Nearest Prototype Classification. *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications. Lecture Notes in Computer Science*, 7441, 300–307.
14. **Jensen, R. & Cornelis, C. (2010).** Fuzzy-rough instance selection. *IEEE World Congress on Computational Intelligence (WCCI 2010),* Barcelona, Spain, 1776–1782.
15. **Bell, D.A. & Guan, J.W. (1998).** Computational methods for rough classification and discovery. *Journal of the American Society for Information Science - Special issue: knowledge discovery and data mining*, 49(5), 403–414.
16. **Santiesteban, Y. & Pons-Porrata, A. (2003).** LEX: A new algorithm to calculate typical testors. Mathematics Sciences Journal, **21**, 31-40.
17. **Suguna, N. & Thanushkodi, K. (2010).** A novel rough set reduct algorithms for medical domains based on bee colony optimization. *Journal of Computing*, 2(6), 49–54.
18. **Banati, H. & Bajaj, M. (2011).** Fire fly based feature selection approach. *International Journal of Computer Science Issues,* 8(4), 473–480.
19. **Camps-Echevarría, L., Llanes-Santiago, O., Silva-Neto, A.J., & Fraga de Campos-Velho, H. (2014).** An Approach to Fault Diagnosis Using Meta-Heuristics: a New Variant of the Differential Evolution Algorithm. *Computación y Sistemas*, 8(4), 473–480.
20. **Bolufé-Röhler, A., Otero, J.M., & Fiol-González, S. (2014).** Traffic Flow Estimation Using Ant Colony Optimization Algorithms. *Computación y Sistemas*, 18(1), 37–50.
21. **Kuncheva, L.I. (1995).** Editing for the k-nearest neighbors rule by a genetic algorithm. *Pattern Recognition Letters*, 16(1995), 809–814.
22. **Kuncheva, L.I. & Bezdek, C.J. (1998).** Nearest prototype classification: clustering, genetic algorithms or random search. *IEEE Transactions on Systems, Man and Cybernetics, Part C: Applications and Reviews*, 28(1), 160–164.
23. **Gagne, C. & Parizeau, M. (2007).** Coevolution of nearest neighbor classifiers. *International Journal of Pattern Recognition and Artificial Intelligence,* 21(5), 921–946.
24. **Siedlecki, W. & Skalansky, J. (1989).** A note on genetic algorithms for large-scale feature selection. *Pattern Recognition Letters*, 10(5), 335–347.
25. **Vafaie, H. & De Jong, K. (1992).** Genetic algorithms as a tool for feature selection in machine learning. *Fourth International Conference on Tools with Artificial Intelligence*, Arlington, VA, 200–203.
26. **Yang, J. & Honavar, V. (1998).** Feature subset selection using a genetic algorithm. *IEEE Intelligent Systems and their Applications*, 13(2), 44–49.
27. **Olveira, L.S., Benahmed, N., Sabourin, R., & Bortolozzi, F. (2001).** Feature subset selection using genetic algorithms for handwritten digit recognition. *14th Brazilian symposium on Computer Graphics and Image Processing*, Florianópolis, Brazil, 362–369.
28. **Sahan, S., Polat, k., Kodaz, H., & Gunes, S. (2007).** A new hybrid method based on fuzzy artificial immune system and k-nn algorithm for breast cancer diagnosis. *Computers in Biology and Medicine*, 37(3), 415–423.
29. **Garain, U. (2008).** Prototype reduction using an artificial immune model. *Pattern analysis & Applications*, 11(3-4), 353–363.
30. **Ahmad, S.S.S. & Pedrycz, W. (2011).** Feature and Instance selection via cooperative PSO. *2011 IEEE International Conference on Systems, Man and Cybernetic,* Anchorage, AK, 2127–2132.

31. **Mohamed, S.S., Youssef, A.M., El-Saadany, E.F., & Salama, M.M.A. (2005).** Artificial life feature selection techniques for prostate cancer diagnosis using TRUS images. *Image Analysis and Recognition. Lecture Notes in Computer Science*, 3656, 903–913.

32. **Polat, K., Sahan, S., Kodaz, H., & Günes, S. (2005).** A new classification method for breast cancer diagnosis: feature selection artificial immune recognition system (FS-AIRS). *Advances in Natural Computation. Lecture Notes in Computer Science*, 3611, 830–838.

33. **Zainal, A., Maarof, M.A., & Shamsuddin, S.M. (2005).** Feature selection using Rough-DPSO in anomaly intrusion detection. *Computational Science and Its Applications-ICCSA 2007. Lecture Notes in Computer Science,* 4705, 512–524.

34. **Bello, R., Nowé, A., Caballero, Y., Gómez, Y.Y., & Vrancx, P. (2005).** Using Ant colony system meta-heuristic and Rough Set Theory for feature selection. *MIC 2005: 6th Metaheuristics International Conference*, Vienna, Austria.

35. **Ruiz-Shulcloper, J. & Abidi, M.A. (2002).** Logical combinatorial pattern recognition: A Review. *Recent Research Developments in Pattern Recognition*, Kerala, India, 133–176.

36. **Garner, S.R. (1995).** WEKA: The Waikato Environment for Knowledge Analysis. *Proceedings of the New Zealand Computer Science Research Students Conference,* Hamilton Beach, New Zealand, 57–64.

37. **Basturk, B. & Karaboga, D. (2006).** An Artificial Bee Colony (ABC) algorithm for numeric function optimization. *IEEE Swarm Intelligence Symposium 2006*, Indianapolis, Indiana, USA.

38. **Karaboga, D. (2005).** *An idea based on honey bee swarm for numerical optimization* (Technical Report TR06). Kayseri, Turquía: Erciyes University.

39. **Karaboga, D. & Basturk, B. (2007).** A powerful and efficient algorithm for numerical function optimization: Artificial Bee Colony (ABC) algorithm. *Journal of Global Optimization*, 39(3), 459–471.

40. **García-Borroto, M. & Ruiz-Shulcloper, J. (2005).** Selecting Prototypes in Mixed Incomplete Data. *Progress in Pattern Recognition, Image Analysis and Applications. Lecture Notes in Computer Science*, 3773, 450–459.

41. **Skalak, D.B. (1994).** Prototype and Feature Selection by Sampling and Random Mutation Hill Climbing Algorithms. *Eleventh International Conference on Machine Learning*. New Brunswick, NJ, USA, 293–301.

42. **Ishibushi, H. & Nakashima, T. (1999).** Evolution of reference sets in nearest neighbor classification. *Simulated Evolution and Learning*. *Lecture Notes on Computer Science*, 1585, 82–89.

43. **Ahn, H. & Kim, K.J. (2009).** Global optimization of case-based reasoning for breast cytology diagnosis. *Expert Systems with Applications*, 36(1), 724–734.

44. **Ahn, H., Kim, K.J., & Han, I. (2007).** A case-based reasoning system with the two-dimensional reduction technique for customer classification. *Expert Systems with Applications*, 32(4), 1011–1019.

45. **Rozsypal, A. & Kubat, M. (2003).** Selecting representative examples and attributes by a genetic algorithm. *Intelligent Data Analysis*, 7(4), 291–304.

46. **Ros, F., Guillaume, S., Pintore, M., & Chrétien, J.R. (2008).** Hybrid genetic algorithm for dual selection. *Pattern Analysis and Applications*, 11(2), 179–198.

47. **Derrac, J., Verbiest, N., García, S., Cornelis, C., & Herrera, F. (2013).** On the use of evolutionary feature selection for improving fuzzy rough set based prototype selection. *Soft Computing*, 17(2), 223–238.

48. **Dasarathy, B.V. & Sánchez, J.S. (2000).** Concurrent Feature and Prototype Selection in the Nearest Neighbor based Decision Process. *4th World Multiconference on Systems, Cybernetics and Informatics*, Orlando, USA, 628–633.

49. **Kittler, J. (1978).** Feature set search algorithms. *Pattern recognition and signal processing* (41–60), The Netherlands: Sijthoff and Noordhoff.

50. **Toussaint, G. (2002).** Proximity Graphs for Nearest Neighbor Decision Rules: Recent Progress*. 34th Symposium on the INTERFACE,* 17–20.

51. **Dasarathy, B.V. (1994).** Minimal consistent set (MCS) identification for optimal nearest neighbor decision systems design. *IEEE Transactions on Systems, Man and Cybernetics*, 24(3), 511–517.

52. **Villuendas-Rey, Y., García-Borroto, M., Medina-Pérez, M.A., & Ruiz-Shulcloper, J. (2006).** Simultaneous features and objects selection for Mixed and Incomplete data. *Progress in Pattern Recognition, Image Analysis and Applications. Lecture Notes on Computer Science*, 4225, 597–605.

Yenny Villuendas-Rey received her M.Sc. degree in Applied Informatics from the University of Ciego de Ávila, in 2007. She is currently a Ph.D. student at the University of Las Villas, Cuba. Since 2005, she has worked as a lecturer at the Engineering Faculty of the University of Ciego de Ávila. She is currently the Head of the Artificial Intelligence research group of the Computer Science Department. Her interests include pattern recognition, case-based reasoning, prototype classifiers, and metaheuristic algorithms.

Maria Matilde Garcia-Lorenzo received her B.Sc. degree in Mathematical Cybernetics from the Universidad Central "Marta Abreu" de Las Villas (UCLV), Santa Clara, Cuba, in 1985 and her Ph.D. in Technical Sciences from UCLV in 1997. She is a full time professor at the Artificial Intelligence Lab of the Center of Studies on Informatics, UCLV, where she is also the General Coordinator of the Master in Computer Science Program (AUIP awarded). Also, she exhibits a long record of academic exchange with many universities in Latin America and Europe. Dr. Garcia has authored 6 books, published over 150 papers in conference proceedings and scientific journals and supervised several Bachelor, Master and Ph.D. theses. She received numerous prestigious awards from the Cuban Academy of Sciences and other renowned scientific societies. Her research interests include neural networks, soft computing, machine learning and pattern recognition.

Article received on 16/08/2013, accepted on 20/03/2014.