

# Support Rough Sets for decision-making

Yenny Villuendas-Rey<sup>1</sup>, Maria M. Garcia-Lorenzo<sup>2</sup>, Rafael Bello<sup>2</sup>

<sup>1</sup>Universidad de Ciego de Avila, Ciego de Avila, Cuba

<sup>2</sup>Universidad Central Marta Abreu de Las Villas, Santa Clara, Cuba

[yenny@informatica.unica.cu](mailto:yenny@informatica.unica.cu); [mmgarcia@uclv.edu.cu](mailto:mmgarcia@uclv.edu.cu); [rbello@uclv.edu.cu](mailto:rbello@uclv.edu.cu)

## Abstract

Case-based reduction is viewed as an important preprocessing step for case based decision-making. In this paper, is introduced a Support Rough Set model to deal with mixed and incomplete data. The Support Rough Set model is used to reduce the case base by using positive and limit regions of decision. The proposed algorithms are compared with some classical techniques. Experimental results show that the proposed methods obtain high accuracy, using only a reduced case base.

**Keywords:** Case based decision-making, Rough Set Theory, data preprocessing

## 1. Introduction

Case Based Reasoning is a problem solving technique that reuses past experience to obtain the solution of a new problem. One of the scenarios of case-based reasoning is when it is needed to take a decision about the label of a new, unknown case, also known as supervised classification.

Among case based reasoning for decision-making, the Nearest Neighbor (NN) [1] classifier is one of the simplest and popular classifiers. On the other hand, the Rough Set Theory has been widely used on decision-making. Pawlak proposed the Rough Set Theory (RST) in 80's [2] and it has had several extensions since then. However, despite the numerous extensions proposed, there still exist several unexplored angles to RST.

In this paper is introduced the Support Rough Set model, as an extension of classical Rough Sets, and it is applied to case based decision-making. Two novel measures of Support Rough Sets are also given in the paper. Four algorithms for case base preprocessing are proposed. The rest of the paper is as follows: section 2 covers some basic concepts about Rough Set Theory and case based decision-making. Proposed Support Rough Sets, and case based preprocessing algorithms based on them are introduced in sections 3 and 4, respectively. Section 5 covers some previous preprocessing algorithms and section 6 presents numerical experiments. Section 7 offers conclusions.

## 2. Rough Set Theory and Case based decision-making

Case based decision-making problems rely on the description of past cases or experiences. In this article, are considered cases described as a  $t$ -tuple of attributes.

Let  $x$  be a case belonging to a case base or universe  $U$ , and described by a set of  $n$  attributes  $A = \{A_1, A_2, \dots, A_n\}$ . Each case has associated a decision label, defining which decision was made in the past, in presence of this case. Formally,  $\forall x \in U$ ,  $d(x)$  denotes the label of the case  $x$ .

Each attribute is defined over a domain  $D_i = \text{dom}(A_i)$ . The domain definition of an attribute will define its set of admissible values. Depending of the nature of this set, an attribute will be Boolean, Numeric  $K$ -valued, among others. If the symbol "?" belongs to the set of admissible values of an attribute, this will denote the presence of missing values, and it will be used when the value of the attribute on a case is unknown. That is, if  $? \in D_i$  can be lack of knowledge of the attribute value on a case.

### 2.1. Basics of Rough Set Theory

Rough Set Theory is based on the assumption that every case  $x$  from a universe  $U$  has associated a certain amount of information (data and knowledge) expressed by means of an attribute set  $A$  that describes the cases [3]. The basic information structure in RST is the Information System. An Information System is a pair  $(U, A)$ . If every case in  $U$  has associated a decision label, the pair  $(U, \{A \cup \{d\}\})$ , where  $d \notin A$  is known as a Decision System.

The decision attribute  $d$  induces a partition of the universe of cases. Let be  $l$  the amount of admissible values of a decision attribute  $d$ , the set  $K = \{K_1, K_2, \dots, K_l\}$  denotes all the possible classes of the universe  $U$ ; and the sets  $K_1, \dots, K_l$  form a collection of equivalence classes, called decision classes, where two objects belong to a same class if they have the same value in the decision attribute.

An attribute  $A_i \in A$  discerns or distinguishes a case  $x$  from another  $y$ , and it is written  $\text{Discern}(A_i, x, y)$ , if and only if  $x(i) \neq y(i)$ , where  $x(i)$  denotes the value of attribute  $i$  in case  $x$ . Each subset of attributes  $B$  de  $A$ ,  $B \subseteq A$ , has associated a binary indiscernibility relation  $R$ , including the set of pairs of cases undistinguished from another according to this relation [3].

$$R = \{(x, y) \in U \times U : x(i) = y(i), \forall A_i \in B\} \quad (1)$$

An indiscernibility relation defined by considering subsets of  $U$  having equal values for a subset of attributes  $B$  from  $A$ ,  $B \subseteq A$ , is an equivalence relation. Lower and upper approximations are the core concepts of the Rough Set Theory. A classical definition of lower and upper approximation was first introduced considering an indiscernibility relation that is an equivalence relation.

Let be  $D=(U, A \cup \{d\})$  a decision system, and  $B \subseteq A$  and  $X \subseteq U$ ,  $B$  defines an equivalence relation, and the subset  $X$  is a concept of the universe  $U$ .  $X$  can be approximated using only the information contained in  $B$  by constructing the  $B$ -lower and  $B$ -upper approximations, denoted by  $B_*X$  and  $B^*X$  respectively, and defined by the expressions (2) and (3), respectively.

$$B_*X = \{x \in U: [x]_B \subseteq X\} \quad (2)$$

$$B^*X = \{x \in U: [x]_B \cap X \neq \emptyset\} \quad (3)$$

where  $[x]_B$  denotes the class of  $x$  according to the indiscernible relation in  $B$ . The cases in  $B_*X$  are with certainty members of  $X$ , while the objects in  $B^*X$  are possible members of  $X$ . Based on lower and upper approximations, the RST offers the Classification Quality measure, defined by the expression (4).

Let be a partition sets  $K = \{K_1, \dots, K_l\}$  of  $U$ , according to the decision attribute (decision labels), where the subsets  $K_i$ , are called decision classes.

$$\gamma_B(K) = \sum_{i=1}^l (|B_*K_i|/|U|) \quad (4)$$

The coefficient  $\gamma_B(K)$  is the percent of cases that can be correctly classified in the classes  $K_1, \dots, K_l$  using only the attributes in  $B$ , where  $l$  is the amount of decision classes.

## 2.2. Extensions to Rough Set Theory

Previous definitions of Rough Sets are based on an equivalence relation  $R$  that defines as indistinguishable the cases having equal attribute values for an attribute subset  $B$ . That is, the cases  $(x, y)$  are indistinguishable according to  $B$  if  $x(i) = y(i)$ , for every attribute  $i$  in  $B$ , where  $x(i)$  denotes the value of the attribute  $i$  in case  $x$ . Defined this way,  $R$  is an equivalence relation.

When the attribute  $B$  contains attributes with numeric domain, an inseparability relation is not applicable. The equivalence relation is too tight in case of continuous domains, which small differences in attribute values may not be significant to analyze its separation. This is very important in the case of numeric attributes, in which small measurement errors may generate these differences in the attribute values.

There are mainly four alternatives to follow when dealing with mixed and incomplete data. First, to analyze separately numeric and categorical attributes, and then merging the decision; second, to discretize numerical attributes; third, codify categorical attributes, and fourth, extend the algorithms and deal with mixed and incomplete descriptions of objects.

In the case of the Rough Set Theory, the most used alternatives are the second (discretize numerical attributes) and the fourth (extend the theory) [4]. In the first case, the original decision system is transformed into another in which classic theory can be applied, and in the last one, the classical approach to Rough Set Theory is extended by accepting that discernible, but very similar cases, can be grouped together in the same equivalence class [5]. In this case, it is needed to use other indiscernibility relations among cases of the universe  $U$ . Thus, it is needed to replace the equivalence relation by a weaker relation, obtaining an extension of classical approach to Rough Set Theory. Some of the reported extensions are studied on [3].

An example of the RST extension is to modify the concept of indiscernible cases, such that very similar, but not identical cases are grouped together according to a similarity relation  $R$ . Similarity relations do not induce partitions of the universe of cases  $U$ , but generate similarity classes for every case  $x \in U$ . The similarity class of  $x$ , according to a similarity relation  $R$ , is denoted by  $R(x)$  and defined as  $R(x) = \{y \in U: yRx\}$

While equivalence relations induce partitions of the universe, similarity relations induce coverings of the universe. A covering of the universe  $U$ , is a family of non-empty subsets which union is equal to the universe. Several researchers [4] have studied RST in approximation spaces forming coverings, several focused on constructing a generalization of RST based on coverings. In this case, the problem is to find an adequate similarity relation for a given decision system.

To solve this problem, Filiberto et al. [4] propose a method based on defining a similarity relation as  $xRy \leftrightarrow F(x, y) \geq \varepsilon$ , where  $x, y \in U$  and  $F(x, y)$  is a similarity function and  $\varepsilon$  is a threshold. In [6] is also proposed to use a threshold for constructing a similarity relation, and dealing with mixed and incomplete data.

However, the problem of finding an adequate threshold for a given decision system is not easily to solve, due to the differences among cases and decision classes. Usually, decision classes have different degrees of dispersions, being in the same decision system compact and disperse decision classes. Defining a unique threshold for every case does not take into account this reality, being a limitation of previous extensions to RST. In the next section is proposed a novel extension, which solves this disadvantage.

## 3. Support Rough Sets

Several extensions to the Rough Set Theory are based on defining a similarity relation that allows obtaining a covering of the universe of cases. The extension proposed in this article is based on two key aspects of the universe of cases: the case based reasoning principle and granularity of information.

Case based reasoning exploits the relation between two kinds of similarities, one defined over space of attribute

descriptions of cases (attribute similarity) and the other defined over the solution space (decision similarity) [7]. Having a correct description of cases, and an adequate definition of the similarity function, a similar case should belong to the same decision class.

Granular Computing includes theories, methodologies, tools, and techniques that use granules (subsets of the universe) to solve problems [8]. The main concept within Granular Computing is the granule, defined as a core, entity, or focal point of knowledge, composite by different cases mutually indistinguishable.

A granulation of the universe consists on decomposing the universe into a family of subsets (granules) that contains all cases of the universe. The granulation depends on a relation  $R$ , defining which cases are together in a granule. In [9] is studied the relation between the conditional granulation (based on attribute values of cases), and decision granulation (based on the decision attribute values).

In this paper, it is considered the decision granulation as the pertinence of cases to decision classes, and the relations  $R1$  and  $R2$  given the condition granulation.

Let  $D=(U,A\cup\{d\})$  be a decision system, where  $U$  denotes the universe of objects, and  $A$  denotes the set of conditional attributes, and let be the similarity relations  $R1$ ,  $R2$ ,  $R3$ , among cases, defined as follows. For every pair of cases  $(x,y)$  in  $U\times U$ .

$$xR1y \leftrightarrow \text{sim}_A(x,y) > \varepsilon_x \quad (5)$$

$$xR2y \leftrightarrow \text{sim}_A(x,y) = \varepsilon_x \quad (6)$$

$$xR1y \leftrightarrow d(x) = d(y) \quad (7)$$

where  $\text{sim}_A$  is a similarity function to compute the similarity degree among pairs of cases of universe  $U$ , considering the attributes in  $A$ ;  $\varepsilon_x$  is a similarity threshold for the case  $x$ . In relation  $R3$ ,  $d(\cdot)$  represents the decision class of cases.

**Definition 1.** Let be  $A$  the attribute set of a decision system  $D$ , and let be  $B \subseteq A$ . the universe of cases with respect to  $B$  is denoted as  $U_B$  and is defined as the set of cases described only by the attributes in  $B$ .

**Definition 2.** Let be  $A$  the set of attributes of a decision system  $D$ ; the set of indexes associated to the set  $A$  is denoted by  $I_A$  and is defined as  $I_A = \{i \in \mathbb{N}; 1 \leq i \leq |A|\}$ .

**Notation 1.** If  $j \in I_A$ ,  $j$  is named the  $j$ -th index of  $A$ , and is denoted as  $I_A(j)$ , that is,  $I_A(j) = j$ ,  $\forall j \in I_A$ .

**Definition 3.** Let be  $A$  the attribute set of a decision system  $D$ , and let be  $B \subseteq A$ . the set of indexes associated to the set  $B$  is denoted as  $I_B$  and is defined as  $I_B = \{i \in I_A; A_{IA(i)} \in B\}$ .

**Lemma 1:** Let be  $A$  the attribute set of a decision system  $D$ , let be  $B \subseteq A$ , and let be  $I_A$  and  $I_B$  the sets of indexes associated to the sets  $A$  and  $B$ , respectively; it is fulfilled that  $I_B \subseteq I_A$ .

**Proof:**

Case 1.  $I_B \neq \emptyset$ . Let be and index  $k \in I_B$ , by definition 3 of  $I_B$ ,  $k \in \{i \in I_A; A_{IA(i)} \in B\}$ , that is,  $k \in I_A$ , so it is concluded that  $I_B \subseteq I_A$ .

Case 2.  $I_B = \emptyset$ . Let's suppose that  $I_B \not\subseteq I_A$ . This means that  $\exists k \in I_B$  such that  $k \notin I_A$ , but this is impossible, because  $I_B = \emptyset$ . So, it is concluded that the thesis is true,  $I_B \subseteq I_A$ . ■

**Definition 4.** Let be  $A$  the set of attributes of a decision system  $D$ ; let be  $I_A$  the set of indexes associated with the set  $A$ , and let be  $i \in I_A$ . Each attribute  $A_i$ , defined over a domain  $D_i = \text{dom}(A_i)$ , has associated a comparison criterion,  $C_i$ . The attribute comparison criterion is defined as a function  $C_i: D_i \times D_i \rightarrow [0,1]$ , where 0 denotes minimum similarity among values of  $A_i$  and 1 denotes maximum similarity.

**Lemma 2:** Let be  $A$  the attribute set of a decision system  $D$ , let be  $B \subseteq A$ . Let be two cases  $x, y \in U_B$ , and let be an index  $k \in I_B$ , then  $0 \leq C_k(x(k), y(k))$ .

**Proof.** By hypothesis and definition 4,  $x(k) \in D_k$ , and  $y(k) \in D_k$ . By lemma 1,  $I_B \subseteq I_A$ , and as  $k \in I_B$ , also  $k \in I_A$ . As the domain of  $C_k$  is  $D_k \times D_k$  is possible to apply, by definition 4,  $C_k$  to the  $k$ -th components of cases  $x$  and  $y$ , denoted as  $x(k)$  and  $y(k)$ , respectively. By definition 4,  $C_k$  is defined over the interval  $[0,1]$ , taking  $C_k(x(k), y(k))$  real values between 0 and 1, including the extremes, so, cannot be lower than 0 or higher than 1, which is expressed by the following inequality:  $0 \leq C_k(x(k), y(k))$ . ■

**Definition 5.** Let be  $A$  the attribute set of a decision system  $D$ , let be  $B \subseteq A$ , and let be  $I_B$  the set of indexes associated to the set  $B$  and let be  $i \in I_B$ . A similarity function among cases  $x, y \in U$ , is defined as a function  $\text{sim}_B: U \times U \rightarrow [0, |B|]$ , such that  $(x, y) \rightarrow \sum_{j \in I_B} C_j(x(j), y(j))$ .

**Definition 6.** The similarity threshold for each case  $(\varepsilon_x)$  is defined as the similarity value of case  $x$  with respect to the most similar case belonging to a different decision class (most similar enemy).

Using relations  $R1$ ,  $R2$  and  $R3$ , are defined the sets  $S_A$ ,  $WS_A$  and  $N$  for every case  $x$  of the universe  $U$ , according to the expressions (8), (9) and (10), where Support ( $S_A(x)$ ), Weak Support ( $WS_A(x)$ ), and Neighbors ( $N(x)$ ) are the sets of cases similar to  $x$ , according to the relations  $R1$ ,  $R2$  and  $R3$ , respectively.

$$S_A(x) = \{y \in U; xR1y\} \quad (8)$$

$$WS_A(x) = \{y \in U; xR2y\} \quad (9)$$

$$N(x) = \{y \in U; xR3y\} \quad (10)$$

Thus, cases belonging to the set  $S_A(x)$ , are the support of  $x$ , due to they are more similar to  $x$  than the most similar cases of different decision classes. Using nearest neighbor as case based reasoning procedure; every of these cases guaranteed the correct classification of case  $x$ . On the other hand, cases belonging to the set  $WS_A(x)$  constitute the weak support of  $x$ , because they have the same similarity value that the most similar enemy of  $x$ .

In the following, some properties of Support Rough Sets are established.

**Lemma 3.** Let be  $A$  the attribute set of a decision system  $D$ , let be  $B \subseteq A$ , and let be  $I_A$  and  $I_B$  the sets of indexes associated to the sets  $A$  and  $B$ , respectively; let be two cases  $x, y \in U_B$ , and let's consider an index  $k \in I_A$ , such that  $k \notin I_B$ . Let's construct the attribute set  $B' = (B \cup A_k) \subseteq A$ , with its set of indexes  $I_{B'} = I_B \cup \{k\}$ . the cases  $x'$  and  $y'$  are formed by the addition of the attribute  $A_k$  to the cases  $x$  and  $y$ , such that  $x', y' \in U_{B'}$ . The similarity value among cases  $x, y \in U_B$  is equal or lower than the similarity value among cases  $x', y' \in U_{B'}$ ,  $\text{sim}_{B'}(x', y') \geq \text{sim}_B(x, y)$ .

**Proof.**  $\text{sim}_{B'}(x', y') = \sum_{j \in I_{B'}} C_j(x(j), y(j))$  by definition 5. As  $I_{B'} = I_B \cup \{k\}$ , this can be rewritten as  $\text{sim}_{B'}(x', y') = \sum_{j \in I_B \cup \{k\}} C_j(x(j), y(j))$ . This can also be rewritten as  $\text{sim}_{B'}(x', y') = \sum_{j \in I_B} C_j(x(j), y(j)) + C_k(x(k), y(k))$ .

Substituting  $\sum_{j \in I_B} C_j(x(j), y(j))$  by  $\text{sim}_B(x, y)$ , is obtained  $\text{sim}_{B'}(x', y') = \text{sim}_B(x, y) + C_k(x(k), y(k))$ . By lemma 2,  $0 \leq C_j(x(j), y(j))$ , so the inequality  $\text{sim}_{B'}(x', y') \geq \text{sim}_B(x, y) + C_k(x(k), y(k))$ . By transitivity,  $\text{sim}_{B'}(x', y') \geq \text{sim}_B(x, y)$ . ■

**Theorem 1.** Let be  $A$  the attribute set of a decision system  $D$ , let be  $B \subseteq A$ , and let be  $I_A$  and  $I_B$  the sets of indexes associated to the sets  $A$  and  $B$ , respectively; let be two cases  $x, y \in U$ , and let's consider an index  $k \in I_A$ , such that  $k \notin I_B$ . Let's construct the attribute set  $B' = (B \cup A_k) \subseteq A$ , with its set of indexes  $I_{B'} = I_B \cup \{k\}$ . the cases  $x'$  and  $y'$  are formed by the addition of the attribute  $A_k$  to the cases  $x$  and  $y$ , such that  $x', y' \in U_{B'}$ . After defined the similarity thresholds  $\epsilon x$  in a decision system of a Support Rough Set, all cases belonging to the set  $S_B(x)$  are maintained in  $S_{B'}(x')$  with the incorporation of condition attributes, that is,  $y \in S_B(x) \rightarrow y' \in S_{B'}(x')$ .

**Proof.** Let be  $y \in S_B(x)$ . By the expressions (8) and (5),  $\text{sim}_B(x, y) > \epsilon x$ .  $\text{sim}_{B'}(x', y') \geq \text{sim}_B(x, y)$ , by lemma 3. By transitivity,  $\text{sim}_{B'}(x', y') > \epsilon x$ . According to the expressions (8) and (5), it means that  $y' \in S_{B'}(x')$ , being  $S_{B'}(x')$  the support set of  $x' \in U_{B'}$ . Considering the expressions  $y \in S_B(x)$  and  $y' \in S_{B'}(x')$ , it is possible to establish that for an arbitrarily case in the support set of  $x$ , according to the attributes in  $B$ ,  $y \in S_B(x)$ , the case  $y' \in S_{B'}(x')$ , which corresponds to the case  $y$  in the universe respect to  $B'$ ,  $U_{B'}$  belongs to the support set of  $x'$  in the universe  $U_B$ . Thus,  $y \in S_B(x) \rightarrow y' \in S_{B'}(x')$ . ■

**Theorem 2.** Let be  $A$  the attribute set of a decision system  $D$ , let be  $B \subseteq A$ , and let be  $I_A$  and  $I_B$  the sets of indexes associated to the sets  $A$  and  $B$ , respectively; let be two cases  $x, y \in U$ , and let's consider an index  $k \in I_A$ , such that  $k \notin I_B$ . Let's construct the attribute set  $B' = (B \cup A_k) \subseteq A$ , with its set of indexes  $I_{B'} = I_B \cup \{k\}$ . the cases  $x'$  and  $y'$  are formed by the addition of the attribute  $A_k$  to the cases  $x$  and  $y$ , such that  $x', y' \in U_{B'}$ . After defined the thresholds  $\epsilon x$  in a decision system, adding new condition attributes cause elements belonging to the set  $WS_B(x)$  upgrade to the set  $S_{B'}(x')$ , or continue in  $WS_{B'}(x')$ , that is,  $y \in WS_B(x) \rightarrow (y \in WS_{B'}(x')) \vee (y \in S_{B'}(x'))$ .

**Proof.** Let be  $y \in WS_B(x)$ . By the expressions (9) and (6),  $\text{sim}_B(x, y) = \epsilon x$ .  $\text{sim}_{B'}(x', y') \geq \text{sim}_B(x, y)$ , by lemma 3. This can be rewritten as  $(\text{sim}_{B'}(x', y') > \text{sim}_B(x, y)) \vee (\text{sim}_{B'}(x', y') = \text{sim}_B(x, y))$ . By transitivity is obtained that  $(\text{sim}_{B'}(x', y') > \epsilon x) \vee (\text{sim}_{B'}(x', y') = \epsilon x)$ . According to the expressions (5) and (8), the first term of the proposition means that  $y' \in S_{B'}(x')$ , being  $S_{B'}(x')$  the support set of  $x' \in U_{B'}$ ; and according to the expressions (9) and (6), the second term means that  $y' \in WS_{B'}(x')$ , being  $WS_{B'}(x')$  the weak support set of  $x' \in U_{B'}$ . Considering the above expressions, it is possible to establish that for an arbitrarily case in the weak support set of  $x$ , according to the attributes in  $B$ ,  $y \in WS_B(x)$ , the case  $y' \in U_{B'}$ , which corresponds to the case  $y$  in the universe respect to  $B'$ ,  $U_{B'}$  belongs to the support set of  $x'$  or belongs to the weak support set of  $x'$  in the universe  $U_B$ . Thus,  $y \in WS_B(x) \rightarrow (y \in WS_{B'}(x')) \vee (y \in S_{B'}(x'))$ . ■

After established the sets  $S_A(x)$ ,  $WS_A(x)$  and  $N(x)$ , the problem consists on defining lower and upper approximations of decision classes. To solve this problem, there are introduced granularity levels for the approximations.

Level 1 (Minimum Granularity) considers that in the lower approximation of a decision class are included cases belonging to this class and having cases in their Support or cases in their Weak Support. Formally, expressions (11) and (12) define the lower and upper approximations of the decision, according to Level 1 of granularity.

$$\text{INF}(K_i) = \{x \in K_i : (S_A(x) \neq \emptyset) \vee (\exists y \in K_i : y \in WS_A(x))\} \quad (11)$$

$$\text{SUP}(K_i) = \{\{x \in K_i\} \cup \{y \notin K_i : y \in WS_A(x)\}\} \quad (12)$$

Cases belonging to lower approximation have at least a neighbor case of its same decision class, with the same or higher similarity value than its most similar enemy. In upper approximation are included all cases belonging to the current decision class  $K_i$ , and cases of other decision classes that are included in the Weak Support ( $WS_A$ ) of cases belonging to  $K_i$ , that is, cases of other classes having the same similarity value than the established threshold.

Another way to define lower and upper approximation is according to level 2, Maximum Granularity. Level 2 considers in the lower approximation only cases having non empty Support, and not included in the Weak Support of cases of different decision classes. Upper approximation is considered as same as in level 1. Expressions (13) and (14) detail the definitions of lower and upper approximations according to level 2.

$$\text{INF}(K_i) = \{x \in K_i : (S_A(x) \neq \emptyset) \wedge (\forall y \notin K_i : x \notin WS_A(y))\} \quad (13)$$

$$\text{SUP}(K_i) = \{\{x \in K_i\} \cup \quad (14)$$

$$y \notin K_i; y \in WS_A(x) \}} \}$$

The concepts associated to granularity levels 1 and 2, are shown graphically taking as example the decision system of figure 1. Figure 1a and 1b show the lower approximation using granularity levels 1 and 2, respectively. Note how cases with only Weak Support are analyzed to be included according to granularity level 1, and how cases with no Support, or belonging to Weak Support of cases from different decision classes are excluded from lower approximation, according to level 2.

As mentioned earlier, one of the most useful measures of the Rough Set Theory is Classification Quality. It measures the goodness of the classification using the Rough Set approximation.

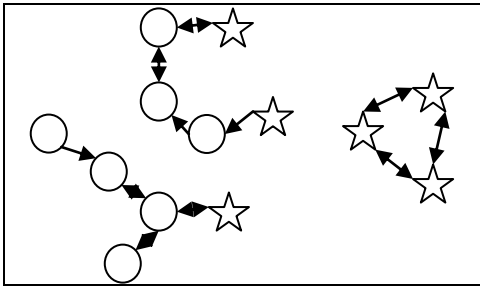


Fig. 1: Decision system formed by two decision classes: circles, and stars. The arrows indicate the direction of the connection of each case with its most similar case.

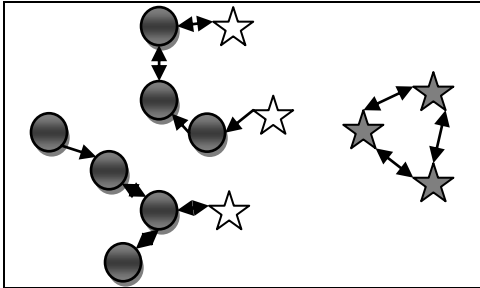


Fig. 1a: Cases belonging to lower approximations according to Level 1 of granularity are filled in gray.

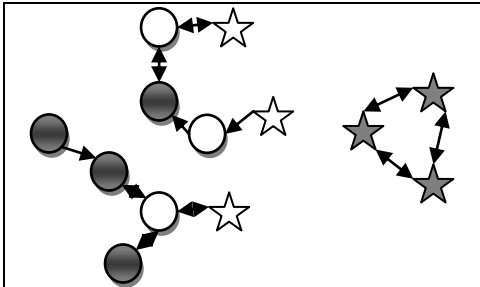


Fig. 1b: Cases belonging to lower approximations according to Level 2 of granularity are filled in gray.

In the paper are also introduced novel measures that can be useful to determine the goodness of certain ap-

proximation of a decision system. Decision Support (15) takes into consideration the relation between the supported cases with respect to the total amount of cases in a decision system.

$$DS_A(D) = OWA_{K_i \in K} (OWA_{x \in K_i} \rho(x)) \quad (15)$$

where  $\rho(x)=1$  if  $S_A(x) \neq \emptyset$ , and  $\rho(x)=0$  elsewhere.

Moreover, Decision Support Ratio (16) takes into consideration the degree in which cases are supported, with respect to the total amount of cases. Both measures can be used to obtain knowledge about the characteristics of decision classes.

Decision Support and Decision Support Ratio are defined using OWA operators [10], which allows using several combinations of them according to the needs present in decision systems.

$$DSR_A(D) = OWA_{K_i \in K} (OWA_{x \in K_i} (|S_A(x)|/|N(x)|)) \quad (16)$$

In this context, determining which cases are relevant and which cases are not, is very important. In this sense, the proposed Support Rough Sets offers some interesting elements. Typical cases usually have high amount of cases in their Support, and less representative cases usually have not. In a similar way, cases belonging to decision boundaries often lack of Support, and merely have cases in their Weak Support.

In addition, both measures can be used to establish the dispersion of decision classes, by analyzing the supports of the cases belonging to them. Moreover, in imbalanced classes, it is possible to determine the relations of the minority class with respect other decision classes.

#### 4. Using Support Rough Sets for case based preprocessing

In this section, four algorithms for case base preprocessing are introduced, based on Support Rough Sets. Two of them are editing methods, ESRS1 and ESRS2 (Editing based on Support Rough Sets, level 1 and 2, respectively), and the other are condensing methods, CSRS1 and CSRS2 (Condensing based on Support Rough Sets, level 1 and 2, respectively). Figure 2 – 3 show the proposed algorithms.

The ESRS1 algorithm will include in the edited case base cases having non-empty Support, and cases having at least one case of its same decision class in its Weak Support. This guarantees each case has at least one case of its decision class with same or higher similarity than its most similar enemy.

Editing based on Support Rough Sets, granularity level 1, ESRS1
Inputs: Case base T, Attribute set B, similarity function $\Delta$ . Output: Edited set E
<ol style="list-style-type: none"> <li>1. <math>E = \phi</math></li> <li>2. Compute thresholds <math>\epsilon x</math> for each <math>x \in T</math></li> <li>3. for each decision class <math>K_i</math>: <ol style="list-style-type: none"> <li>3.1. for each case <math>x \in K_i</math> <ol style="list-style-type: none"> <li>Compute sets <math>S_A(x)</math> and <math>WS_A(x)</math> according to similarity relations R1 and R2</li> </ol> </li> <li>3.2. Compute the lower approximation of decision class <math>K_i</math> at level 1, as <math>INF(K_i)</math></li> </ol> </li> <li>4. <math>E = E \cup INF(K_i)</math></li> <li>5. Return E</li> </ol>

Fig. 2: Editing algorithm based on Support Rough Sets, granularity level 1.

ESRS2 algorithm is more demanding, because it uses level 2 of granularity, and only includes in the edited set cases having non-empty Support, and not included in the Weak Support of cases of different decision classes.

On the other hand, if it is wanted to preserve decision boundaries it is needed to use other strategies. The condensing algorithms CSRS1 and CSRS2 are based on limit region of decision, and therefore they preserve decision boundaries.

In the Rough Set Theory, the limit region of a decision class is defined as the intersection of upper and lower approximations of the decision class, and the limit region of a decision system is defined as the union of limits regions of each decision classes. Formally, for a decision class  $K_i$ , its limit region is defined as  $LIM(K_i) = SUP(K_i) - INF(K_i)$ . In addition, the limit region of a decision system is given as  $LIM(D) = \cup_{K_i \in K} LIM(K_i)$ .

However, the limit region of a class contains only boundary cases, so, to obtain a good representation of the decision class, the algorithms select representative cases. To structure a class in subclasses, Compact sets have been very useful [11]. Compact sets are the connected components of a Maximum Similarity Graph (MSG).

A Maximum Similarity Graph is a directed graph that connects each case with its most similar neighbors. Formally, let be  $G=(X,\theta)$  a MSG for a set of cases X, with arcs  $\theta$ . In this graph, two cases  $x,y \in X$  form an arc  $(x,y) \in \theta$  if  $\max_{t \in X} \{\Delta(x,y)\}$ , where  $\Delta(x,y)$  is a similarity function. For a more formal definition of Compact set, please refer to [12].

Compact sets do not assume any properties of data, and do not need any parameter for their construction, except of the similarity function to compare two objects. They also handle mixed and incomplete data. As mention, the decision class is structured using Compact sets.

Then, for each Compact set, the algorithm selects as representative case the case more similar to every case in the compact set. Formally, the representative case is selected as in expression (17).

$$r = \operatorname{argmax}_{x \in K_i} \{ \sum_{y \in K_i} \Delta(x,y) \} \quad (17)$$

Same as CSRS1, CSRS2 obtains a condensed set, but using the limit region of the decision according to granularity level 2.

Condensing based on Support Rough Sets, granularity level 1, CSRS1
Inputs: Case base T, Attribute set B, similarity function $\Delta$ . Output: Condensed set C
<ol style="list-style-type: none"> <li>1. <math>C = \phi</math></li> <li>2. Compute thresholds <math>\epsilon x</math> for each <math>x \in T</math></li> <li>3. for each decision class <math>K_i</math>: <ol style="list-style-type: none"> <li>Compute limit region of the class, according to level 1, as <math>LIM(K_i) = SUP(K_i) - INF(K_i)</math></li> <li><math>C = C \cup LIM(K_i)</math></li> <li>Obtain compact sets <math>CS(K_i)</math></li> <li>for each compact set in <math>CS(K_i)</math>, compute the representative case r using equation (17) and add it to C.</li> </ol> </li> <li>4. Return C</li> </ol>

Fig. 3: Condensing algorithm based on Support Rough Sets, granularity level 1.

CSRS2 includes in the condensed set the cases having no Support, and those included in the Weak Support of cases of different decision classes. CSRS\_2, same as CSRS\_1, guarantees the representation in the condensed set of the classes and the inclusion of boundary cases.

## 5. Previous works on case base preprocessing for decision-making

Case based decision-making incorporates strategies for storage and retrieval of information of the case base to determine the decision to make in presence of new, unseen cases.

The case base is the core of case based decision-making, and its quality is crucial for the entire reasoning process. Rough Set Theory has been successfully used for case base preprocessing, by selecting relevant features [13, 14], relevant cases [15-17], and more recently, for selecting features and cases [18].

The methods for case based preprocessing are divided in editing methods and condensing methods [19]. Editing methods aim at smoothing decision boundaries among classes, by removing noisy or mislabeled cases. The first method for this end is the ENN, proposed by Wilson in 1972 [20]. Despite its age, the ENN method is considered as one of the best preprocessing methods, because it is simple and has very good experimental results.

Another technique used for editing case bases are the proximity graphs, particularly the Relative Neighborhood Graph. The Relative Neighborhood Editing (RNE), by Toussaint, is based on such graphs [21]. Recently, Garcia-Borroto et al. proposed the MSEditB method for case based preprocessing. We will refer the MSEditB method

as MSB in experimental comparisons. The method is based on removing cases connected with cases from different decision classes in a Maximum Similarity Graph [22].

On the other hand, condensing methods aim at selecting representative cases only, and preserving decision boundaries. The first condensing method was CNN, proposed by Hart in 1968 [23]. CNN has served as inspiration for numerous proposals, and has many extensions and variants, such as GCNN [24]. Among other condensing algorithms, it is possible to mention GBR1, based on Gray Relational Graphs, introduced in 2006 by Huang [25]. Another recently proposed algorithm for data condensing, also based on Maximum Similarity Graphs, is the CSESupport method, introduced in 2009 by Garcia-Borroto et al. [26]. We will refer to the CSESupport method as CSES in experimental comparisons. This method aims at obtaining a consistent set with minimum cardinality.

## 6. Experimental Results

This section compares the performance of proposed editing and condensing algorithms for case base preprocessing, over 20 case bases from the Machine Learning repository of the University of California at Irvine [27]. Table 1 shows the description of them, including attribute count (Att.), case count (Cases), class count (Classes) and the error of the NN classifier (NN) without preprocessing.

Table 1. Description of case bases.

Case bases	Att.	Cases	Classes	NN
autos	26	205	6	0.29
breast-c	9	289	2	0.35
breast-w	9	699	2	0.05
colic	22	368	2	0.20
credit-a	15	690	2	0.18
cylinder	40	512	2	0.23
dermat.	34	366	6	0.06
diabetes	8	768	2	0.29
ecoli	7	336	8	0.20
hayes	5	160	4	0.31
heart-c	13	303	5	0.23
heart-h	13	294	5	0.24
iris	4	150	3	0.05
labor	14	57	2	0.14
thyroid	5	215	3	0.03
tic-tac	9	958	2	0.22
vehicle	18	946	4	0.31
vowel	12	990	11	0.01
wine	13	178	3	0.04
zoo	17	101	7	0.03

As similarity function, there was used  $sim(x,y)=\sum_{i=1}^m Ci(x(i),y(i))$ . The attribute comparison criterion  $Ci$  compares attribute values, depending if they are numeric, nominal or incomplete. For incomplete attributes  $Ci(x(i),y(i))=0$ , for numeric attributes,

$Ci(x(i),y(i)) = (|x(i)-y(i)|)/(maxi-mini)$ , and for nominal attributes,  $Ci(x(i),y(i))=1$  if  $x(i)=y(i)$ , and 0 otherwise.

To determine the performance of proposed algorithms, it was compared the ESRS1 and ESRS2 algorithms with respect to editing methods MSB [22], and RNE [21]. It was also compared the performance of CSRS1 and CSRS2 with respect to condensing methods CSES [28] and GBR1[25].

It was used 10-fold cross validation over every case base. Then, each algorithm was applied and the results were averaged. It was used the Nearest Neighbor (NN) rule as case based decision-making method. As quality measures, there were computed the average by class error of classification, and the case retention rate, for each algorithm.

Tables 2 – 7 show the results comparing the performance of proposed methods with respect to others. As shown in table 2, the method with low classifier errors is RNE; followed by the proposed ESRS2 and by other editing algorithms. In addition, ESRS1 algorithm is also comparable according to classifier error with other editing methods.

Table 3 shows that the proposed ESRS2 method obtains the best results according to case retention in 13 of 20 databases.

To determine if significant differences exist in the performance of the algorithms, it was used the Wilcoxon's test, with a 95% of significance. The null hypothesis was that the compared algorithms had the same performance.

Table 4 shows the results of the Wilcoxon's test, comparing the performance of ESRS1 and ESRS2 with respect other algorithms, according to classifier error and case retention. Each cell contains the probability given by the test.

Table 2. Error of editing algorithms.

Case bases	RNE	MSB	ESRS1	ESRS2
autos	0.15	0.40	0.42	0.38
breast-c	0.35	0.40	0.43	0.39
breast-w	0.41	0.04	0.04	0.03
colic	0.04	0.21	0.23	0.23
credit-a	0.22	0.15	0.15	0.14
cylinder	0.14	0.32	0.33	0.30
dermat.	0.30	0.06	0.06	0.06
diabetes	0.06	0.30	0.30	0.29
ecoli	0.30	0.21	0.23	0.22
hayes	0.21	0.34	0.40	0.26
heart-c	0.46	0.19	0.19	0.18
heart-h	0.19	0.22	0.22	0.21
iris	0.21	0.03	0.03	0.03
labor	0.04	0.09	0.19	0.09
thyroid	0.10	0.08	0.08	0.09
tic-tac	0.10	0.24	0.24	0.18
vehicle	0.26	0.32	0.32	0.32
vowel	0.31	0.03	0.03	0.03
wine	0.02	0.04	0.04	0.02
Zoo	0.04	0.08	0.06	0.07
Times	11	5	4	8

The test concludes that ESRS2 is better than the MSB method and equal to RNE, according to classifier accuracy. In addition, it keeps fewer cases than both algorithms. On the other hand, ESRS1 ties with RNE in both classifier error and cases retention, but it loses with respect to MSB. These results suggest that granularity level 2 is a very good tool for editing databases, obtaining good classifier accuracy with fewer cases.

Tables 5 and 6, show the results of condensing algorithms, according to the classifier error and case retention, respectively.

Table 3. Retention of editing algorithms.

Case bases	RNE	MSB	ESRS1	ESRS2
autos	0.87	0.71	0.75	0.65
breast-c	0.82	0.70	0.82	0.32
breast-w	0.63	0.94	0.95	0.93
colic	0.56	0.76	0.80	0.58
credit-a	0.82	0.80	0.81	0.76
cylinder	0.86	0.75	0.77	0.66
dermat.	0.82	0.93	0.94	0.93
diabetes	0.97	0.68	0.71	0.63
ecoli	0.78	0.79	0.80	0.76
hayes	0.87	0.62	0.89	0.47
heart-c	0.27	0.74	0.76	0.70
heart-h	0.85	0.76	0.78	0.73
iris	0.85	0.94	0.96	0.93
labor	0.96	0.83	0.86	0.77
thyroid	0.91	0.96	0.97	0.94
tic-tac	0.98	0.99	1.00	0.16
vehicle	0.05	0.67	0.70	0.62
vowel	0.79	0.99	0.99	0.98
wine	0.98	0.93	0.95	0.90
zoo	0.96	0.96	0.97	0.87
Times Best	7	0	0	13

Table 4. Wilcoxon's test on editing algorithms. In bold significant results favoring proposed methods and in italics results rejecting proposed methods.

	Error		Retention	
	RNE	MSB	RNE	MSB
ESRS1 vs	0.79	<i>0.04</i>	0.68	<i>0.00</i>
ESRS2 vs	0.80	<b>0.03</b>	<b>0.04</b>	<b>0.00</b>

Condensing algorithms usually obtain reduced training sets by increasing the classifier error. As shown in table 5, the proposed CSRS1 and CSRS2 have slightly higher classifier errors than unprocessed case base, with accuracy drops lower than 15%. With respect to CSES algorithm, the proposed methods have lower error rates in average.

cy drops lower than 15%. With respect to CSES algorithm, the proposed methods have lower error rates in average.

Table 5. Error of condensing algorithms.

Case bases	GBR1	CSES	CSRS1	CSRS2
autos	0.29	0.29	0.27	0.27
breast-c	0.50	0.44	0.52	0.47
breast-w	0.10	0.10	0.11	0.09
colic	0.23	0.23	0.35	0.26
credit-a	0.21	0.26	0.22	0.21
cylinder	0.27	0.29	0.28	0.27
dermat.	0.08	0.12	0.11	0.12
diabetes	0.35	0.38	0.37	0.36
ecoli	0.26	0.36	0.34	0.33
hayes	0.40	0.35	0.32	0.40
heart-c	0.21	0.26	0.24	0.24
heart-h	0.16	0.33	0.29	0.29
iris	0.05	0.12	0.05	0.05
labor	0.12	0.10	0.07	0.13
thyroid	0.06	0.12	0.07	0.07
tic-tac	0.24	0.26	0.37	0.26
vehicle	0.31	0.33	0.31	0.33
vowel	0.01	0.04	0.04	0.03
wine	0.07	0.08	0.07	0.07
zoo	0.08	0.05	0.07	0.03
Times Best	13	1	5	7

Table 6 shows the results of case retention rates of the condensing algorithms. As shown, proposed methods have higher retention rates than CSES. However, they are able to preserve decision boundaries, in the reduced set of cases.

Again, Wilcoxon's test was used to determine if there significant differences exist in the performance of the algorithms (Table 7). The test does not found significant differences between proposed methods and GBR1 and CSES methods, according to the classifier error.



Table 6. Retention of condensing algorithms.

Case bases	GBR1	CSES	CSRS1	CSRS2
autos	0.90	0.43	0.57	0.66
breast-c	1.00	0.63	0.21	0.71
breast-w	0.19	0.10	0.13	0.15
colic	0.19	0.29	0.26	0.48
credit-a	0.57	0.30	0.45	0.50
cylinder	0.66	0.37	0.52	0.63
dermat.	0.88	0.13	0.26	0.27
diabetes	0.72	0.41	0.52	0.60
ecoli	0.44	0.29	0.44	0.48
hayes	0.47	0.62	0.32	0.74
heart-c	0.43	0.34	0.49	0.54
heart-h	0.04	0.30	0.46	0.51
iris	0.03	0.12	0.31	0.34
labor	0.34	0.16	0.34	0.43
thyroid	0.74	0.09	0.32	0.35
tic-tac	0.93	0.84	0.01	0.84
vehicle	0.62	0.42	0.57	0.65
vowel	0.92	0.20	0.32	0.33
wine	0.97	0.11	0.27	0.32
zoo	0.65	0.12	0.17	0.28
Times Best	3	14	3	0

Table 7. Wilcoxon's test on condensing algorithms. In bold significant results favoring proposed methods and in italics results rejecting proposed methods.

	Error		Retention	
	GBR1	CSES	GBR1	CSES
CSRS1 vs	0.05	0.23	<b>0.01</b>	<b>0.08</b>
CSRS2 vs	0.11	0.16	0.31	<i>0.00</i>

According to instance retention, CSRS1 method outperforms both GBR1 and CSES, while CSRS2 method ties with GBR1 and loses with respect to CSES. The experiments show that on condensing algorithms, keeping only limit cases preserves decision boundaries, and gives a good representation of the decision class.

Using Support Rough Sets with granularity level 1 in condensing methods are obtained slightly better results than using granularity level 2, but the Wilcoxon's test does not found significant differences with respect to other methods. It is important to mention that the proposed editing methods obtain good classifier accuracy, with average retention of about 65%-85% of cases.

In addition, condensing algorithms are comparable according to the classifier accuracy with previously reported method, but using fewer amount of cases, obtaining an average retention of about 35%-55% of

cases. The above results show using Support Rough Sets for case based preprocessing enhances the case base, and improves decision-making procedures.

## 7. Conclusions

Reducing redundant or irrelevant cases can improve decision-making performance in most cases and decrease cost of storage and retrieval of cases. Classical Rough Set Theory has been widely applied for case selection and data preprocessing. However, this approach only deals with nominal data. In this work, there are introduced Support Rough Sets as an extension of classical Rough Sets.

This extension allows dealing with mixed and incomplete data. Support Rough Sets are also used in four algorithms for case base preprocessing. In addition, both Similarity Support and Similarity Support Ratio measures can be used to characterize decision systems, as well as to select relevant cases and attributes.

The experimental results show that the proposed algorithms can be used to deal with both categorical and numerical attributes directly, without discretization and they are able to find small and effective subsets of cases.

## References

- [1] T. M. Cover and P. E. Hart, "Nearest Neighbor pattern classification," *IEEE Transactions on Information Theory*, vol. 13, pp. 21-27, 1967.
- [2] Z. Pawlak, "Rough Sets," *International Journal of Information & Computer Sciences*, vol. 11, pp. 341-356, 1982.
- [3] Z. Pawlak and A. Skowron, "Rough sets: Some extensions," *Information Sciences*, vol. 177, pp. 28-40, 2007.
- [4] Y. Filiberto-Cabrera, R. Bello, Y. Caballero, and R. Larrúa, "A method to build similarity relations into extended Rough Set Theory," in *10th International Conference on Intelligent Systems and Applications ISDA 2010*, Cairo, Egypt, 2010.
- [5] R. Slowinski and D. Vanderpooten, "Similarity relation as a basis for rough approximations," in *Advances in Machine Intelligence & Soft-Computing*, Duke University Press ed. vol. IV, P. P. Wang, Ed. Durham, 1997, pp. 17-33.
- [6] Q. Hu, D. Yu, J. Liu, and C. Wu, "Neighborhood rough sets based heterogeneous feature selection," *Information Sciences*, vol. 178, pp. 3577-3594, 2008.
- [7] R. L. Lopez and E. Armengol, "Machine learning from examples: Inductive and Lazy methods," *Data & Knowledge Engineering*, vol. 25, pp. 99-123, 1998.
- [8] W. Pedrycz, A. Skowron, and V. Kreinovich, *Handbook of Granular Computing*. West Sussex: Wiley, 2008.

- [9] B. Chen, et al., "Granular rough theory: A representation semantics oriented theory of roughness," *Applied Soft Computing*, vol. 9, pp. 786-805, 2009.
- [10] R. Fuller, "OWA Operators in Decision Making," in *TUCS general publications, No. 3*, C. Carlsson, Ed.: Turku Center for Computer Science, Abo, 1996, pp. 85-103.
- [11] M. García-Borroto and J. Ruiz-Shulcloper, "Selecting Prototypes in Mixed Incomplete Data," *Lecture Notes in Computer Science*, vol. 3773, pp. 450-459, 2005.
- [12] J. Ruiz-Shulcloper and M. A. Abidi, "Logical combinatorial pattern recognition: A Review," in *Recent Research Developments in Pattern Recognition*, S. G. Pandalai, Ed. USA: Transworld Research Networks, 2002, pp. 133-176.
- [13] H. Banati and M. Bajaj, "Fire fly based feature selection approach," *International Journal of Computer Science Issues*, vol. 8, pp. 473-480, 2011.
- [14] H. Fan and Y. Zhong, "A Rough Set approach to feature selection based on Wasp Swarm optimization," *Journal of Computational Information Systems*, vol. 8, pp. 1037-1045, 2012.
- [15] Y. Caballero, R. Bello, Y. Salgado, and M. M. García, "A method to edit training set based on rough sets," *International Journal of Computational Intelligence Research*, vol. 3, pp. 219-229, 2007.
- [16] R. Jensen and C. Cornelis, "Fuzzy-Rough instance selection," in *IEEE World Congress on Computational Intelligence*, Barcelona, Spain, 2010, pp. 1776-1782.
- [17] Y. Villuendas-Rey, Y. Caballero-Mota, and M. M. García-Lorenzo, "Using Rough Sets and Maximum Similarity Graphs for Nearest Prototype Classification," in *17th Iberoamerican Congress on Pattern Recognition, CIARP 2012*. vol. LNCS 7441 Buenos Aires, Argentina: Springer-Verlag, 2012, pp. 300-307.
- [18] Y. Villuendas-Rey, Y. Caballero-Mota, and M. M. García-Lorenzo, "Intelligent feature and instance selection to improve Nearest Neighbors classifiers," in *11th Mexican International Conference on Artificial Intelligence, MICAI 2012*. vol. LNCS: Springer-Verlag, 2012.
- [19] S. García, J. Derrac, J. R. Cano, and F. Herrera, "Prototype Selection for Nearest Neighbor Classification: Taxonomy and Empirical Study," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, pp. 417-435, 2012.
- [20] D. L. Wilson, "Asymptotic properties of nearest neighbor rules using edited data," *IEEE Transactions on systems, man and cybernetics*, vol. SMC-2, pp. 408-421, 1972.
- [21] G. T. Toussaint, "Proximity Graphs for Nearest Neighbor Decision Rules: Recent Progress," in *34 Symposium on Computing and Statistics INTERFACE-2002*, Montreal, Canada, 2002, pp. 1-20.
- [22] M. García-Borroto, Y. Villuendas-Rey, J. A. Carrasco-Ochoa, and J. F. Martínez Trinidad, "Using Maximum Similarity Graphs to edit nearest neighbor classifiers," *Lecture Notes on Computer Science*, vol. 5856, pp. 489-496, 2009.
- [23] P. E. Hart, "The condensed nearest neighbor rule," *IEEE Trans. on Information Theory*, vol. 14, pp. 515-516, 1968.
- [24] C. H. Chou, B. A. Kuo, and F. Cheng, "The Generalized Condensed Nearest Neighbor rule as a data reduction technique," in *18th International Conference on Pattern Recognition*, 2006, pp. 556-559.
- [25] C.-C. Huang, "A novel gray-based reduced NN classification method," *Pattern Recognition*, vol. 39, pp. 1979-1986, 2006.
- [26] M. García-Borroto, Y. Villuendas-Rey, J. A. Carrasco-Ochoa, and J. F. Martínez Trinidad, "Finding Small Consistent Subset for the Nearest Neighbor Classifier Based on Support Graphs," *Lecture Notes on Computer Science*, vol. 5856, pp. 465-472, 2009.
- [27] C. J. Merz and P. M. Murphy, "UCI Repository of Machine Learning Databases," University of California at Irvine, Department of Information and Computer Science, 1998.
- [28] J. Hua, W. D. Tembe, and E. R. Dougherty, "Performance of feature-selection methods in the classification of high-dimension data," *Pattern Recognition*, vol. 42, pp. 409-424, 2009.