Intelligent Feature and Instance Selection to Improve Nearest Neighbor Classifiers

Yenny Villuendas-Rey[1,3], Yailé Caballero-Mota[2], and María Matilde García-Lorenzo[3]

[1] Computer Science Department, University of Ciego de Ávila, Carr. a Morón km 9 ½, Cuba
yennyv@informatica.unica.cu

[2] Computer Science Department, University of Camagüey, Circunv. Norte km 3 ½, Cuba
yaile.caballero@reduc.edu.cu

[3] Computer Science Department, University of Las Villas, Carr. a Camajuaní, km 5 ½, Cuba
mmgarcia@uclv.edu.cu

Abstract. Feature and instance selection before classification is a very important task, which can lead to big improvements in both classifier accuracy and classifier speed. However, few papers consider the simultaneous or combined instance and feature selection for Nearest Neighbor classifiers in a deterministic way. This paper proposes a novel deterministic feature and instance selection algorithm, which uses the recently introduced Minimum Neighborhood Rough Sets as basis for the selection process. The algorithm relies on a metadata computation to guide instance selection. The proposed algorithm deals with mixed and incomplete data and arbitrarily dissimilarity functions. Numerical experiments over repository databases were carried out to compare the proposal with respect to previous methods and to the classifier using the original sample. These experiments show the proposal has a good performance according to classifier accuracy and instance and feature reduction.

Keywords: instance selection, object selection, rough sets, nearest neighbor.

1 Introduction

Increasing the efficiency of Case Based Reasoning techniques constitutes a significant research area in Artificial Intelligence. One of the key topics in this research is case base preprocessing. It may include instance selection or generation, feature selection, and simultaneous or combined feature and instance selection. In the latter, the algorithms select both features and instances, obtaining a highly reduced and accurate case base. Previous work done by Kuncheva and Jain [1] show that simultaneous selection of features and instances leads to better results than sequential selection. The quality of the case base is important to every supervised classifier, and Nearest Neighbor (NN) is one of the most affected by it; because it stores the case base and compares every new case with those stored, having a time and memory costs increasing with the dimensions of the case base. There are several methods to improve NN classifiers through simultaneous or combined feature and instance selection, having some drawbacks such as a stochastic nature, high computational

I. Batyrshin and M. González Mendoza (Eds.): MICAI 2012, Part I, LNAI 7629, pp. 27–38, 2013.

cost, insufficient noise filtering, and inability to deal with imbalanced case bases. This paper introduces a novel deterministic method to improve NN classifier by selecting features and instances, which makes this process better than other combined methods. The main contributions of the proposal are:

1. It has strong theoretic basis, because it uses extended Rough Set Theory [2] and structuralizations of the Logical Combinatorial Approach to Pattern Recognition [3].
2. It is deterministic, and deals with mixed as well as imbalanced data.
3. It uses metadata to determine the condensing or editing strategy to follow in instance selection procedure.
4. It obtains high data reduction, maintaining the original classifier error.

The organization of the contribution is as follows: the next section covers some general concepts about extended Rough Set Theory and structuralizations of the Logical Combinatorial Approach to Pattern Recognition. Section 3 explains the proposed approach to feature and instance selection, and Section 4 contains several numerical experiments to determine the performance of the proposal with respect to other feature and object selection methods. Section 5 gives the conclusions and future works.

2 Maximum Similarity Graphs and Rough Sets

2.1 Maximum Similarity Graphs

The Logical Combinatorial approach to Pattern Recognition has several data structuralization procedures [3], which have their basis on Maximum Similarity Graph computation. A Maximum Similarity Graph (MSG) is a directed graph such that it connects each instance with all of its most similar instances. More properly, let be $G = (X, \theta)$ a MSG for a set of instances X, with arcs θ. In this graph, two instances $x_i, x_j \in X$ form an arc $(x_i, x_j) \in \theta$ if and only if $\max_{x \in X}\{sim(x_i, x)\} = sim(x_i, x_j)$, where $sim(x_i, x_j)$ is a similarity function. If there are several instances with maximum similarity with respect to an instance x (ties), the MSG establishes an arc between x and each of its more similar instances. Each connected component of a MSG is a Compact Set (CS). Compact sets guaranteed that the most similar example of each instance belong to the same compact of the instance. It is usual to construct a MSG using $sim(x_i, x_j) = 1 - \Delta(x_i, x_j)$, where $\Delta(x_i, x_j)$ is a dissimilarity function.

Maximum Similarity Graphs are the basis for several prototype selection methods, such as [4-6], and offers several advantages to data analysis. They do not assume any properties of data and do not need any parameter for their construction, except the similarity function to compare two instances. They also handle mixed as well as incomplete data.

2.2 Minimum Rough Sets as Extended Rough Sets

Rough Set Theory (RST) was proposed by Pawlak in 1982 [7]. It assumes that each object x of an universe U has related a certain amount of information, and the attributes or features that describe the object express it. In RST, the basic structure of information is the Information System. An Information System is a pair S= (U, F), where U is a non-empty finite set of objects called the Universe and $F = \{f_1, f_2, \ldots, f_n\}$ is a non-empty finite set of features. The classification data form a Decision System, which is any Information System such that $DS = F \cup \{d\}$, where $d \notin F$ is the decision feature. The decision feature d induces a partition of the universe U. Let be the sets $Y_i = \{x \in U: x(d) = i\}$, $\{Y_1, \ldots, Y_b\}$ is a collection of equivalence classes, named decision classes, where the objects belong to the same class if and only if they have the same value at the decision attribute d. Each subset B of F, B⊆F, has associated a binary indiscernible relation denoted by R, which is the set of object pairs which are indiscernible according to the relation [2]. An equivalence relation is an indiscernible relation defined by forming subsets of objects of U having the same values of a subset of features B of F, B⊆F.

When dealing with continuous attributes, an equivalence relation as defined previously is not appropriate, since some close values may be similar, but discernible. An extension of the classical RST is to modify the concept of indiscernible objects, such that similar objects according to a similarity relation R are together in the same class. The similarity relations generate similarity classes, for each object $x \in U$. The recently introduced Minimum Neighborhood Rough Sets [8] defines the similarity relation using Maximum Similarity Graph concepts. Two objects are similar (neighbors) if they form an arc in a Maximum Similarity Graph, that is, the Neighborhood of an object is $N_B(x_i) = \{x_j \mid (x_i, x_j) \in \theta\}$. Let be $\mathrm{Y_i} \in \mathrm{Y}$ a decision class, its positive region is as following:

$$POS_B(\mathrm{Y_i}) = x_i \left| x_i \in X, \forall_{x_j \in N_B(x_i)}, x_i(\mathrm{d}) = x_j(\mathrm{d}) = \mathrm{i} \right. \tag{1}$$

Therefore, objects with pure neighborhood will form the positive region of the decision classes. The limit region of the decision contains objects with neighbors of different classes (equation 2). This generalization allows handling mixed data, and using specific similarity functions, without any threshold definition.

$$LIM_B(\mathrm{Y_i}) = x_i \left| x_i \in X, \exists_{x_j \in N_B(x_i)}, x_i(\mathrm{d}) \neq x_j(\mathrm{d}) \right. \tag{2}$$

As shown, extended Rough Set Theory has several advantages to data analysis, such as it does not need any external information; no assumptions about data are necessary, and it is suitable for analyzing both quantitative and qualitative features.

3 Intelligent Feature and Instance Selection

As stated before, combined feature and instance selection (FIS) algorithms obtain better results than sequential selection [1]. This may be due to these algorithms use the information of the entire case base (CB) in order to obtain a reduced one, while in sequential selection the second method only has access to the results of the first one.

According to the nature of the selection process, FIS algorithms are stochastic or deterministic. Among stochastic algorithms, there has been and extensive use of Genetic Algorithms [9], Swarm Intelligence techniques [10], Cooperative Co-evolution [11] and Hybrid methods [12]; however, stochastic methods are beyond the scope of this paper. Much little work exists on deterministic FIS methods for mixed data. The first algorithm for this purpose is the proposed by Dasarathy in [13], which uses a wrapper selection strategy. Another deterministic algorithms are SOFSA (Simultaneous Object and Feature Selection Algorithm) [14] and TCCS (Testor and Compact set based Combined Selection) [15], which use a combined selection strategy. Although SOFSA and TCCS deal with mixed data, they do not filter noise and have a sub-matrixes fusion strategy based only on a sub-matrixes sort procedure. To overcome these drawbacks, this section introduces the proposed algorithm for feature and instance selection. The IFIS (Intelligent Feature and Instance Selection) algorithm has four steps (figure 1). Step one addresses initial noise filtering or data condensation, depending of a metadata designed to determine what action to take. Step 2 obtains several sub-matrixes using the support set (section 3.2) obtained using the entire training set and Step 3 sorts them according to classifier accuracy. Step 4 merges the sub-matrixes using a fusion procedure.

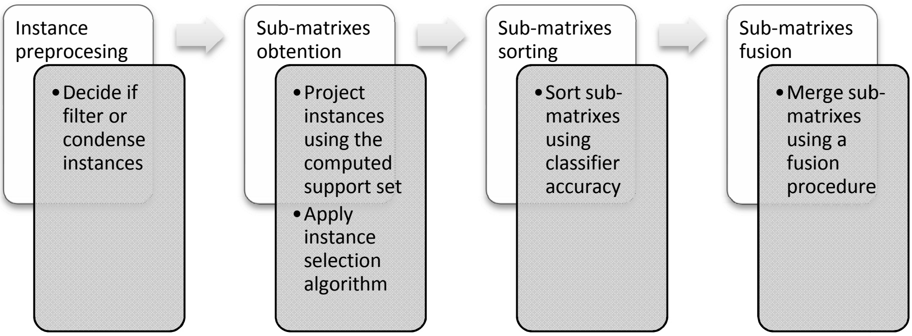

Fig. 1. Proposed IFIS algorithm

3.1 Filtering Noise or Condensing Data

Among instance selection methods, there are error - based editing methods and condensing methods. Error – based editing typically delete noisy or mislabeled instances in class boundary regions, while condensing methods keep border instances and delete the redundant ones. Error – based editing tries to smooth class boundaries and decreasing Bayes error; but in some cases, they may delete an entire class. On the other hand, condensing may keep many instances, decreasing classifier accuracy. Pioneer work of Dasarathy in 2000 [16] tried to exploit the benefit of both error – based editing and condensing strategies, and minimize their weakness through a synergic exploitation. However, the question of when to apply each strategy is still open. In the authors' opinion, there are two key factors in deciding the performance of error – based editing and condensing methods: class overlapping and class imbalance. In case of imbalanced data, if the minority class has overlapping with another class, error – based editing may delete the entire minority class.

When dealing with well-separated classes, condensing methods obtain very good results, and error – based editing do not offer a good instance reduction. On the other hand, when classes have a certain degree of overlapping, and are balanced, error - based editing obtain very good performance. On the contrary, if exist a high degree of class overlapping, and balanced data, it is not clear which strategy perform better. Considering this, the IFIS algorithm uses a metadata to guide the initial instance-preprocessing step. The proposed metadata computes Class Overlapping (CO) as the maximum amount of instances of a class having a heterogeneous arc in a Maximum Similarity Graph (equation 3), and Imbalance Ratio (IR) as the ratio between the amount of instances of majority and minority class, respectively.

$$CO = \max_{i} \frac{|\{x \in T, x(d) = i | \exists \theta(x,y), x(d) \neq y(d)\}|}{|\{x \in T, x(d) = i\}|} \tag{3}$$

The metadata has the form of decision rules (figure 2) and plays a key role in given and "intelligent" decision in preprocessing stage of IFIS algorithm. If CO is "Low", IFIS applies first an editing method. On the contrary, if CO is "High", IFIS applies a condensing method. In addition, if CO is medium, IFIS analyzes IR. If IR is "Low", IFIS applies first an editing method, and if IR is "High", IFIS applies a condensing method. The values of "Low", "Medium" and "High" where obtained by discretizing CO and IR.

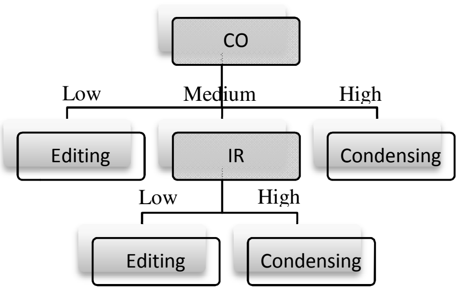

Fig. 2. Metadata rules to guide instance preprocessing. The leaves of the tree indicate the strategy to follow (editing or condensing).

The procedure for instance preprocessing using error - based editing is as follows: it computes the positive region of each decision class, according to the Minimum Neighborhood Rough Sets and deletes the object not present in any positive region. In a different way, the procedure for instance preprocessing using condensing computes the limit region of the decision, and deletes the objects not in the limit region.

Procedure for instance preprocessing
Inputs: Training set T, Attribute set B, Dissimilarity Δ. Outputs: Preprocessed training set P
1. Obtain a Maximum Similarity Graph, $G = (T, \theta)$ of the objects in T 2. Compute Metadata and decide the strategy (editing or condensing) according to the rules. 3. **If** (strategy = editing) Compute the positive region of the Decision System as $POS_B(\mathrm{Y}) = \bigcup_{\mathrm{i}} POS_B(\mathrm{Y_i})$,

where $POS_B(\mathrm{Y_i}) = x_i \left| x_i \in X, \forall_{x_j \in N_B(x_i)},\ x_i(\mathrm{d}) = x_j(\mathrm{d}) = \mathrm{Y_i} \right.$

Remove the objects not included in the positive region of the Decision System, as $P = T - POS_B(\mathrm{Y})$

Else

Compute the limit region of the Decision System as $LIM_B(\mathrm{Y}) = \bigcup_{\mathrm{i}} LIM_B(\mathrm{Y_i})$, where $LIM_B(\mathrm{Y_i}) = x_i \left| x_i \in X, \exists_{x_j \in N_B(x_i)},\ x_i(\mathrm{d}) \neq x_j(\mathrm{d}) \right.$

Remove the objects not included in the limit region of the Decision System, as $P = T - LIM_B(\mathrm{Y})$

4. Return P

3.2 Obtaining Sub-matrixes

In early 70's Zhuravlev and Nikiforov [17] introduce the idea of using a feature support set system to classification. A support set system is a set $S = \{F_1, \dots, F_k\}$, where each set F_i is a feature set. Having a support set system allows using different subspaces to project instances, to improve overall classifier accuracy, as in ALVOT classifiers [3]. An example of a support set system is the set of all reducts (or typical testors) of a training data. The concept of reduct in Rough Set Theory attains to an irreducible set of features B such that using an equivalence relation R, the set of indiscernible objects using all features $IND(F)$ is equal to the set of indiscernible objects using only the features in B, $IND(B)$; that is, it preserves the partition of the universe [7]. In addition, the concept of typical testor in Logical Combinatorial approach to Pattern Recognition, first proposed by Zhuravlev in the past century [18], attains to a set of features such that it does not confuse instances of different classes and are irreducible. As shown, the concepts of reduct and typical testor, although described in different scenarios, reference the same feature sets.

To obtain sub-matrixes, IFIS needs a feature support set system and an instance selection algorithm. IFIS computes a support set system by using the entire training set, without any preprocessing. Then, the procedure projects the preprocessed instances using each feature set in the support set system, and then applies the instance selection method to each projection (figure 3).

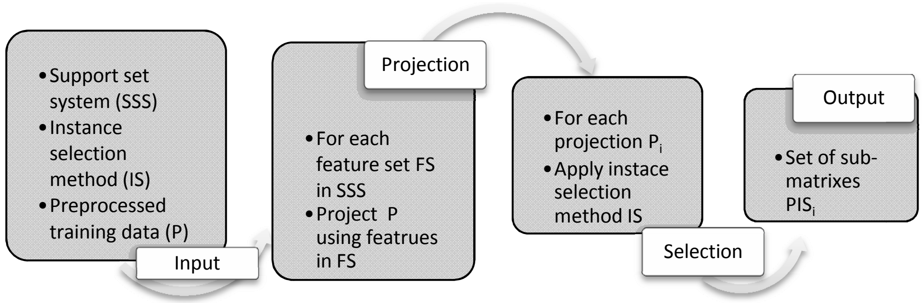

Fig. 3. Procedure to obtain sub-matrixes

The procedure will obtain as many sub-matrixes as feature sets in the support set system. The authors consider that using an instance selection method that obtain a good representation of the entire training set will lead to better results than using error-based editing or condensing methods with low object retention rates. Section 4 discusses in detail the influence of the instance selection method in IFIS performance.

3.3 Sorting and Fusing Sub-matrixes

The IFIS algorithm obtains several sub-matrixes in the previous step. Then, it associates to each sub-matrix a fitness value that determines the quality of the sub-matrix. The fitness value may correspond to classifier accuracy, or to a Rough Set Theory measure such as Classification quality [2]. The sorting procedure sorts the sub-matrixes descendant or ascendant, depending of the fitness function. Therefore, the procedure guarantees best sub-matrixes being first in the resulting list. In this paper, IFIS uses classifier error of the training set as fitness function. Usually, the best sub-matrix obtained by the sorting procedure is worse than the original training set. Therefore, the fusion procedure (figure 4) merges it with other sub-matrixes to improve classifier accuracy.

The procedure uses a greedy approach; each iteration finds the available sub-matrix that decreases the most the classifier error. The process continues until no sub-matrixes are available or the classifier error is lower than original. The fusing procedure does not resembles the original training set, because sub-matrixes are obtained using only the instances in the preprocessed training set (section 3.1) and the feature set of the support set system (section 3.2).

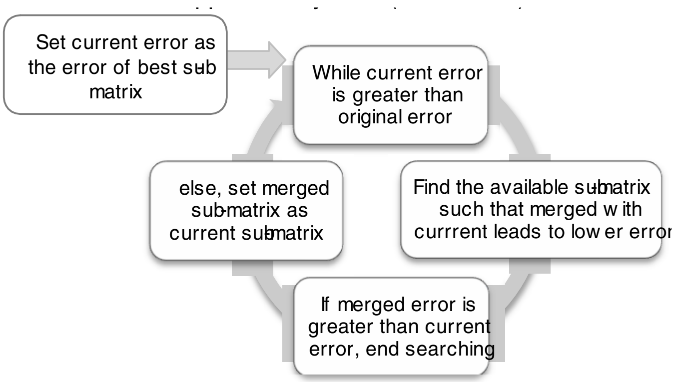

Fig. 4. Procedure to fuse sub-matrixes

Due to its greedy approach, it is reasonable that the fusion strategy of IFIS will obtain better results in object retention that the one of TCCS. Section 4 addresses this topic in detail.

4 Experimental Results

4.1 Experimental Setup

This section addresses some numerical experiments to test the performance of the IFIS algorithm. The selected twenty six databases are from the Machine Learning Repository of University of California at Irvine [19]. Table 1 gives the description of them. Marked with * databases having missing values.

Table 1. Description of the databases used in numerical experiments

Databases	*Attributes (Categorical -Numerical)*	*Obj.*	*IR*	*Databases*	*Attributes (Categorical -Numerical)*	*Obj.*	*IR*
anneal*	29-9	798	86.51	heart-h*	7-6	294	1.77
autos*	10-16	205	23.13	hepatitis*	13-6	155	3.87
breast-c*	9-0	289	2.37	iris	0-4	150	1.09
breast-w	0-9	699	1.90	labor	6-8	57	1.86
car	6-0	1728	18.69	lymph	15-3	148	47.55
colic*	15-7	368	1.73	new-thyroid			5.01
credit-a*	9-6	690	1.25	tae	2-3	151	1.09
credit-g	13-7	1000	2.35	tic-tac-toe			1.89
diabetes	0-8	768	1.87	vehicle	0-18	946	1.10
ecoli	2-5	336	93.05	vote*	16-0	435	1.59
glass	0-8	214	8.48	vowel	3-9	990	1.12
hayes-roth			2.14	wine*	0-13	178	1.47
heart-c*	7-6	303	1.20	zoo	16-1	101	10.46

The fist experiment studies the influence of using positive and limit region in IFIS preprocessing. Then, subsection 4.2 explores different instance selection methods in IFIS performance. Next subsection compares the fusion strategies of TCC and IFIS and subsection 4.4 studies the performance of IFIS using different dissimilarity functions. The first three experiments use as object dissimilarity the HEOM (equation 4), and the later also uses the HVDM (equation 5), both proposed by Wilson and Martínez [20]. max_a and min_a are the maximum and minimum values of attribute a, respectively. C is the amount of classes, $N_{a,x}$ is the amount of objects having value x at attribute a, $N_{a,x,c}$ is the amount of objects of class c having value x at attribute a, and q is a constant, usually 1 or 2.

$$HEOM(x,y) = \sqrt{\sum_{a=1}^{m} d_a(x_a,y_a)}\,,\ d_a = \begin{cases} 1 \\ overlap(x_a,y_a), \\ diff(x_a,y_a) \end{cases} \tag{4}$$

$$overlap(x_a,y_a) = \begin{cases} 0 & if\ x_a = y_a \\ 1 & elsewhere \end{cases},\ diff(x_a,y_a) = |x_a - y_a| / (max_a - min_a)$$

$$HVDM(x,y) = \sqrt{\sum_{a=1}^{m} vdm_a(x_a, y_a)}\,, vdm_a = \sum_{c=1}^{C} \left| \frac{N_{a,x,c}}{N_{a,x}} - \frac{N_{a,y,c}}{N_{a,y}} \right|^q \quad (5)$$

4.2 Influence of the Positive and Limit Regions in IFIS

Among deterministic feature and instance selection algorithms, TCCS [15] has very good performance. This section compares the usefulness of the positive and limit region on IFIS algorithm, using TCCS and original classifier as base algorithms. To test only the influence of the preprocessing stage, in this experiment IFIS use the same sorting and fusion strategy of TCCS. Also use the same support set system and instance selection method (typical testors and CSE [4], respectively).

Cross validation is a standard procedure to compare the performance of supervised classification algorithms; therefore, all experiments use 10-fold cross validation and average results. To statistically determine if the differences in performance were significant, Demsar [21] recommends using Wilcoxon test, also employed in all experiments with a 95% of confidence. Table 2 gives the results of the Wilcoxon's test comparing IFIS using the preprocessing step (IFIS-P) with respect to Original classifier (Orig.) and TCCS. In this experiment, IFIS uses the same parameters and procedures than TCCS. Each column show the probability of the Wilcoxon's test, and the times the proposal wins, losses or ties with respect other. In bold the times the test found significant differences. Feature retention of both algorithms has no significant differences, because they use the same support set system.

Table 2. Results of the Wilcoxon test comparing IFIS preprocessing

Pair	*Error*		*Instance Retention*	
	wins-losses-ties	prob.	wins-losses-ties	prob.
Orig. vs TCCS	22-4-0	**0.000**	0-26-0	**0.000**
Orig. vs IFIS-P	15-11-0	0.082	0-26-0	**0.000**
TCCS vs IFIS-P	12-14-0	0.675	1-25-0	**0.000**

The results show that the preprocessing procedure of IFIS maintains original classifier accuracy, having no significant differences with respect to the original classifier. On the contrary, TCCS loses 22 times with respect to the original classifier error. However, IFIS-P has no differences in error with respect TCCS. According to object retention, both TCCS and IFIS-P obtain a reduced set of instances, but IFIS-P achieves much reduction than TCCS, being better on 25 databases.

4.3 Influence of the Instance Selection Method in IFIS

As stated before, the instance selection algorithm may influence the results of IFIS. This experiment compares the performance of IFIS using CSE [4] (IFIS-CSE) and CSESupport [5] (IFIS-CS) as instance selection algorithms. Both CSE and CSESupport rely on Maximum Similarity Graph computation. CSE tries to preserve the structure of data, using the sub-class consistency property [4], while CSESupport looks for a minimum consistent set. Table 3 gives the results according to classifier error and instance retention, using Wilcoxon test. Although the results show no difference in classifier error using CSE and CSESupport on IFIS (probability value greater than 0.05), the results of both methods with respect to classifier accuracy differ.

Table 3. Results of the Wicoxon test comparing different instance selection methods in IFIS

Pair	*Error*		*Instance Retention*	
	wins-loses-ties	prob.	wins-loses-ties	prob.
Orig. vs IFIS-CSE	16-10-0	0.082	0-26-0	**0.000**
Orig. vs IFIS-CS	17-9-0	**0.015**	0-26-0	**0.000**
IFIS-CSE vs IFIS-CS	14-10-2	0.189	0-23-3	**0.000**

IFIS-CSE has lower error than IFIS-CS, having a higher probability value compared with respect to the original classifier. On the other hand, IFIS-CS keeps much less objects than IFIS-CSE (due to the significant difference of both algorithms, that favors IFIS-CS). The experiment shows that IFIS is dependant of the instance selection method. The authors recommend using structure-preserving algorithms such as CSE to preserve original accuracy, and using high-condensing methods such as CSESupport to obtain as much instance reduction as possible.

4.4 Influence of the Fusion Strategy in IFIS

IFIS introduces a novel fusion strategy using a greedy approach. This subsection compares the utility of the novel strategy (IFIS-N) by comparing it with the fusion strategy of TCCS (IFIS-TC). Both algorithms use the same preprocessing step, as well as support set systems, sorting and instance selection algorithm. Table 4 gives the results according to classifier accuracy and instance retention, by means of Wilcoxon test.

Table 4. Results of the Wicoxon test comparing IFIS fusion

Pair	*Error*		*Instance Retention*	
	wins-loses-ties	prob.	wins-loses-ties	prob.
Orig. vs IFIS-N	16-10-0	**0.055**	0-26-0	**0.000**
Orig. vs IFIS-TC	16-10-0	0.082	0-26-0	**0.000**
IFIS-TC vs IFIS-N	7-3-16	0.139	0-13-13	**0.001**

The above results show the novel fusion strategy maintains classifier accuracy, tying with original classifier and IFIS-TC. In addition, it leads to a much-reduced training set, being significantly better than the fusion strategy of TCCS.

4.5 Influence of the Dissimilarity Function in IFIS

Finally, this section compares the performance of IFIS using HEOM and HVDM dissimilarities. Table 5 shows the results. Our proposal does not closely depend of the dissimilarity function used. It obtains the best results according to object reduction, being significantly better than TCCS with HEOM and HVDM dissimilarities. According to classifier error, IFIS obtains very good results. It ties with the original classifier and with TCCS.

Table 5. Results of the Wicoxon test comparing IFIS with different disimilarities

Pair	*Error*		*Instance Retention*	
	wins-loses-ties	prob.	wins-loses-ties	prob.
Orig. vs IFIS-HEOM	15-11-0	0.082	0-26-0	0.000
TCCS vs IFIS-HEOM	12-14-0	0.675	1-25-0	0.000
Orig. vs IFIS-HVDM	17-9-0	0.218	0-26-0	0.000
TCCS vs IFIS-HVDM	11-15-0	0.603	0-26-0	0.000

It is important to mention that IFIS maintains classifier accuracy using a very reduced training set. The above results show that using positive or limit regions of a Minimum Neighborhood Rough Set, leads to better results than directly use training instances. Also, the sorting and fusion strategies introduced by IFIS, obtain better results in instance retention and classifier accuracy than previuos methods.

5 Conclusions

Nearest Prototype Classification offers several advantages to Nearest Neighbor classifiers. However, it suffers dealing with mixed data is still a challenge for prototype selection algorithms. The proposed IFIS algorithm for combined feature and instance selection uses extended Rough Set Theory and structuralizations of the Logical Combinatorial Approach to Pattern Recognition to instance preprocessing, deciding the best editing or condensing strategy through a metadata computation. IFIS is deterministic, and deals with mixed as well as imbalanced data. The experimental results show the sorting and fusion strategies introduced by IFIS, obtain better results in instance retention and classifier accuracy than previuos methods, with high data reduction and maintaining the original classifier error.

References

1. Kuncheva, L.I., Jain, L.C.: Nearest neighbor classifier: Simultaneous editing and feature selection. Pattern Recognition Letters 20, 1149–1156 (1999)
2. Pawlak, Z., Skowron, A.: Rough sets: Some extensions. Information Sciences 177, 28–40 (2007)
3. Ruiz-Shulcloper, J., Abidi, M.A.: Logical combinatorial pattern recognition: A Review. In: Pandalai, S.G. (ed.) Recent Research Developments in Pattern Recognition. Transword Research Networks, USA, pp. 133–176 (2002)

4. García-Borroto, M., Ruiz-Shulcloper, J.: Selecting Prototypes in Mixed Incomplete Data. In: Sanfeliu, A., Cortés, M.L. (eds.) CIARP 2005. LNCS, vol. 3773, pp. 450–459. Springer, Heidelberg (2005)
5. García-Borroto, M., Villuendas-Rey, Y., Carrasco-Ochoa, J.A., Martínez-Trinidad, J.F.: Finding Small Consistent Subset for the Nearest Neighbor Classifier Based on Support Graphs. In: Bayro-Corrochano, E., Eklundh, J.-O. (eds.) CIARP 2009. LNCS, vol. 5856, pp. 465–472. Springer, Heidelberg (2009)
6. García-Borroto, M., Villuendas-Rey, Y., Carrasco-Ochoa, J.A., Martínez-Trinidad, J.F.: Using Maximum Similarity Graphs to Edit Nearest Neighbor Classifiers. In: Bayro-Corrochano, E., Eklundh, J.-O. (eds.) CIARP 2009. LNCS, vol. 5856, pp. 489–496. Springer, Heidelberg (2009)
7. Pawlak, Z.: Rough Sets. International Journal of Information & Computer Sciences 11, 341–356 (1982)
8. Villuendas-Rey, Y., Caballero-Mota, Y., García-Lorenzo, M.M.: Using Rough Sets and Maximum Similarity Graphs for Nearest Prototype Classification. In: Alvarez, L., Mejail, M., Gomez, L., Jacobo, J. (eds.) CIARP 2012. LNCS, vol. 7441, pp. 300–307. Springer, Heidelberg (2012)
9. Ahn, H., Kim, K.J., Han, I.: A case-based reasoning system with the two-dimensional reduction technique for customer classification. Expert Systems with Applications: An International Journal 32, 1011–1019 (2007)
10. Sakinah, S., Ahmad, S., Pedrycz, W.: Feature and Instance selection via cooperative PSO. In: IEEE International Conference on Systems, Man and Cybernetic, pp. 2127–2132. IEEE Publishing (2011)
11. Derrac, J., García, S., Herrera, F.: IFS-CoCo in the Landscape Contest: Description and Results. In: Ünay, D., Çataltepe, Z., Aksoy, S. (eds.) ICPR 2010. LNCS, vol. 6388, pp. 56–65. Springer, Heidelberg (2010)
12. Derrac, J., Cornelis, C., Gaecía, S., Herrera, F.: Enhancing evolutionary instance selection algorithms by means of fuzzy rough set based feature selection. Information Sciences 186, 73–92 (2012)
13. Dasarathy, B.V.: Concurrent Feature and Prototype Selection in the Nearest Neighbor Decision Process. In: 4th World Multiconference on Systemics, Cybernetics and Informatics, Orlando, USA, vol. VII, pp. 628–633 (2000)
14. Villuendas-Rey, Y., García-Borroto, M., Medina-Pérez, M.A., Ruiz-Shulcloper, J.: Simultaneous Features and Objects Selection for Mixed and Incomplete Data. In: Martínez-Trinidad, J.F., Carrasco Ochoa, J.A., Kittler, J. (eds.) CIARP 2006. LNCS, vol. 4225, pp. 597–605. Springer, Heidelberg (2006)
15. Villuendas-Rey, Y., García-Borroto, M., Ruiz-Shulcloper, J.: Selecting Features and Objects for Mixed and Incomplete Data. In: Ruiz-Shulcloper, J., Kropatsch, W.G. (eds.) CIARP 2008. LNCS, vol. 5197, pp. 381–388. Springer, Heidelberg (2008)
16. Dasarathy, B.V., Sanchez, J.S., Townsend, S.: Nearest Neighbour Editing and Condensing Tools - Synergy Exploitation. Pattern Analysis & Applications 3, 19–30 (2000)
17. Zhuravlev, Y.I., Nikiforov, V.V.: Recognition algorithms based on voting calculation. Journal Kibernetika 3, 1–11 (1971)
18. Lazo-Cortés, M., Ruiz-Shulcloper, J., Alba-Cabrera, E.: An overview of the evolution of the concept of testor. Pattern Recognition 34, 753–762 (2001)
19. Merz, C.J., Murphy, P.M.: UCI Repository of Machine Learning Databases. University of California at Irvine, Department of Information and Computer Science, Irvine (1998)
20. Wilson, R.D., Martinez, T.R.: Improved Heterogeneous Distance Functions. Journal of Artificial Intelligence Research 6, 1–34 (1997)
21. Demsar, J.: Statistical comparison of classifiers over multiple datasets. The Journal of Machine Learning Research 7, 1–30 (2006)