Genetic Feature Selection for Low Quality Data

Luciano Sánchez
Univ. Oviedo, D. Informática
33071 Gijón, Asturias
luciano@uniovi.es

José R. Villar
Univ. Oviedo, D. Informática
33071 Gijón, Asturias
villarjose@uniovi.es

Inés Couso
Univ. Oviedo, D. Estadística
33071 Oviedo, Asturias
couso@uniovi.es

Abstract

Acquiring precise data is expensive. Hence, there is interest in the development of computer algorithms that make full use of imprecise data, even though these new algorithms may be more complex and require more resources. In particular, most Genetic Fuzzy Systems (GFS). accept crisp inputs, but changes can be effected to them so that rules are obtained from vague data. Furthermore, the rule generation is only one stage in the design of a model. If a GFS uses vague data, we also need to preprocess this low quality data before the learning can take place, but there are very few algorithms capable of selecting vague features or detecting redundant imprecise instances, for example.

In this work we propose a wrapper-type evolutionary feature selection algorithm, able to use incomplete and imprecise data. In the context of Genetic Learning of Fuzzy Rule-based Classifier Systems (FRBCS), we have applied it to remove unnecessary features of fuzzy discretized data.

Keywords: Genetic Fuzzy Systems, Vague Data, Feaure Selection.

1 Introduction

The selection of a subset of features for classification problems can be solved either with wrapper or filter methods. Wrappers consider that the classification algorithm is a black box, used by the search algorithm to evaluate each feature subset. Instead, filter methods are independent of the classifier and select features based on properties that good feature sets are supposed to have. Filter methods can produce wrong results, because they do not have into account the learning algorithm. In contrast, the main problem with wrappers is the computing time.

If the learning algorithm is fast, binary coded genetic algorithms can be used to search subsets of features with good results [1]. Otherwise, the genetic algorithm can be combined with a different classifier which is faster to learn, but then some of the advantages of the wrapper algorithms over filters are lost. Both approaches have also been combined. For instance, in [3, 16, 2] genetic search and filters are combined. A comprehensive review of the use of GAs in feature selection algorithms can be found in [2].

1.1 Vague data and FRBCS

The preceding approaches are suitable for precise, numerical data. This is not always the case. If a high accuracy is needed, then acquiring precise data is expensive. Hence, many real-world datasets are coarsely measured. Also, missing values, or incomplete inputs, can corrupt otherwise precise data.

From a theoretical point of view, it is widely accepted that defuzzifying or removing the low quality parts of the data is worse than carrying these imprecise magnitudes through the computer algorithm, and taking into account the dispersion they produce in the output. We have advocated the use of fuzzy data in GFS [4], and proposed the use of fuzzy fitness functions for extracting rules from vague data in classification [9, 10] and regression problems [11].

We have also studied before how to preprocess imprecise databases. Indeed, there are many recent works dealing with feature selection procedures that use fuzzy techniques [12, 13] or are designed to be used in combination with fuzzy systems [14, 15]. However, to the best of our knowledge, the only paper where the feature selection of fuzzified continuous data has been studied is [6], where a filter method, based on a similarity function, was used. In previous works, we have also proposed a filter-type evolutionary algorithm, that uses a mutual information measure to perform feature selection from vague data [7][8]. Alternatively, in this work we will propose a wrapper-type evolutionary feature selection algorithm that can also use vague data. The new proposal is based on our own extension to the fuzzy case of the k-NN classification algorithm, thus in this sense our algorithm can be regarded as a fuzzy generalization of the SSGA algorithm [16].

This paper is organized as follows: In Section 2 we introduce our extension of the k-NN algorithm for fuzzy data. In Section 3 we describe the genetic search of the set of features, and in Section 4 we include some benchmarks. The paper finishes with the concluding remarks and the future work.

2 Use of the nearest neighbor inductor in a wrapper algorithm with uncertain data

The most frequent use of the term "fuzzy k-NN" is described in [19]. In this work, a membership value for each crisp example in the training dataset is introduced, and the class of the object is assigned to the class with higher certainty, in a procedure similar to a statistical kernel classifier. Many publications extend this definition or apply it to practical problems [20, 21]. Even though some of these extensions use a fuzzy set for defining the class of an object [22], to the best of our knowledge a k-NN algorithm making use of imprecise data has not yet been published.

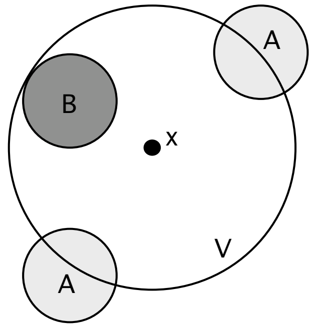

Figure 1: Interval-valued data: The smallest ball that contains for sure the nearest object B has non-null intersection with two other objects of class A. Therefore, $P(C_A|x) \in [0, 2/3]$ and $P(C_B|x) \in [1/3, 1]$ and we can not label x (i.e., we assign the set of labels $\{C_A, C_B\}$ to x).

2.1 An extended definition of the k-NN criterion for fuzzy data

From an statistical point of view, the k-NN rule can be derived from the Bayes rule. Let us restrict ourselves, without loss of generality, to the two-classes problem and let us be given a sample of size N, where N_A elements are in class A and $N_B = N - N_A$ in B. We want to classify the point x. The minimum error classifier is defined as

$$P(C_A|x) \gtrless P(C_B|x). \qquad (1)$$

For estimating these probabilities, we rewrite the expression (1) first,

$$P(C_A|x) = \frac{f(x|C_A)p(C_A)}{f(x)} \qquad (2)$$

then estimate both density functions from the sample. Let V be the smallest ball that contains k objects of the sample. Let n_A,

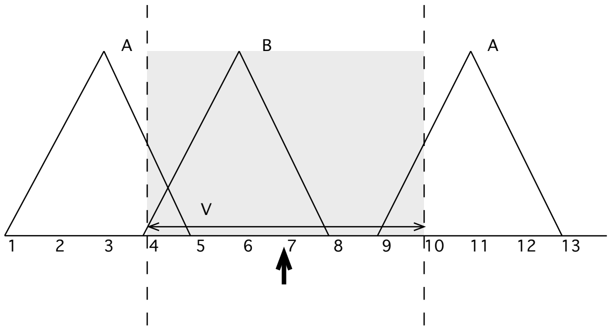

Figure 2: Fuzzy data: The smallest volume centered in 7 that completely contains the object of class B is the interval $V = [4, 10]$. For α-cuts lower than 0.5, the estimations of $P(C_A|x)$ and $P(C_B|x)$ intersect. For $\alpha > 0.5$, $P(C_B|x) > P(C_A|x)$. Therefore, the class of x is the fuzzy set $1/C_B + 0.5/C_A$.

$n_B = k - n_A$ be the number of objects of the classes A and B in V. Then, if V is small enough,

$$\frac{f(x|C_A)p(C_A)}{f(x)} \approx \frac{\frac{n_A}{N_A V}\frac{N_A}{N}}{\frac{k}{NV}} = \frac{n_A}{k}. \quad (3)$$

Hence, eq. (1) reduces to the k-NN rule: we label x as the class that appears the most in the smallest ball, centered in x, that contains k objects in the sample.

Let us suppose now that we can not precisely observe x, but an interval that contains it. For example, consider the situation in Figure 1: the smallest volume V that contains one element of the sample also intersects two other objects, but it does not completely contain them. In this example, the application of eq. (1) requires deciding whether $[0, 2/3] \gtreqless [1/3, 1]$, and we do not have information enough to know the response, thus we will label the example x with the whole set $\{C_A, C_B\}$.

The fuzzy case is an extension of the interval case. Let us consider the data displayed in Figure 2. We have a one-dimensional problem, where we want to label the point $x = 7$, according to the nearest neighbor rule. We have three imprecisely measured objects surrounding x: two of them, the triangular fuzzy numbers $(1; 3; 5)$ and $(9; 11; 13)$, belong to class A. A third one, $(4; 6; 8)$, belong to class B. The smallest volume, centered in 7, that contains one element of the sample, is the interval $[4, 10]$. Now observe that each α-cut of these three sets forms an interval-valued classification problem. For $\alpha \leq 0.5$, V intersects with the three objects. For levels greater than 0.5, the only object in V is that of class B. Therefore, our knowledge about the class of the point x is given by the fuzzy set

$$1/C_B + 0.5/C_A. \quad (4)$$

Summarizing, in case we are given a sample $(S_1, C_1), \ldots, (S_N, C_N)$ of classified objects, where the measurements taken over each object are crisp sets S_i and the class of each object is an element of the set $\{C_1, \ldots, C_M\}$, we define first the values $\hat{P}_*(C|x)$ and $\hat{P}^*(C|x)$ as the minimum and maximum of the set

$$\left\{ \frac{\sum_{C_i=C} a_i}{\sum a_i} : a_i \in \left\{ \begin{array}{ll} \{0\} & S_i \cap V = \emptyset \\ \{1\} & S_i \subseteq V \\ \{0,1\} & \text{else} \end{array} \right. \right\} \quad (5)$$

where V is the smallest sphere, centered in x, that completely contains k objects of the sample. Our extended k-NN rule assigns to each point x a subset C of $\{C_1, \ldots, C_M\}$, defined as follows:

$$C(x) = \{C_j : \hat{P}_*(C_i|x) \leq \hat{P}^*(C_j|x), \quad i \neq j\}. \quad (6)$$

In case the measurements are fuzzy sets X_i, the extended k-NN rule assigns to the point x a fuzzy set of classes. Let us define the level cut α of the sample as the interval-valued dataset $([X_1]_\alpha, C_1), \ldots, ([X_N]_\alpha, C_N)$. If we applied the preceding rule for classifying x on the basis of a level cut α of the sample, we would obtain a (crisp) set of classes $C_\alpha(x)$. We propose that the class of x is the fuzzy subset of $\{C_1, \ldots, C_M\}$ defined by the membership functions

$$\mu_{C_j} = \sup\{\alpha : C_j \in C_\alpha(x)\}. \quad (7)$$

It is emphasized that, for classifying either a crisp or a fuzzy set instead of a point (and we want to do this, if we are selecting features in vague datasets), we have to make sure that the volume V completely contains k elements of the sample but also that it contains the whole area being classified. However, we have a certain freedom in the definition of some of the properties of V (for instance, that of V being centered in the point being classified) as soon as V is small enough for the approximation in eq. (3) making sense.

2.2 Symbolic data

The expression (2) holds when x is a vector of real numbers. Instead, when x is an element in a finite space, we have to assume some degree of smoothness in $p(x|C_A)$ in order to estimate its value at a point which does not appear in the sample. Usually, we admit that eq. (3) still holds when the volume V is defined wrt a certain distance. The most common distance is the count of features that match, although there are more complex approaches based on distance tables between features [5].

However, our particular problem is simpler than that. We are mostly interested in a rather common situation in FRBCS, that of numerical variables that are transformed into fuzzy subsets of the set of linguistic labels by means of a Ruspini's fuzzy partition. For example, the fuzzification stage can convert a numerical value of 45 degrees into a fuzzy subset like {0.0/COLD + 0.2/WARM+0.8/HOT}. Rule based-systems could also manage subsets like {0.1/COLD + 0.3/WARM + 0.9/HOT} or {0.5/COLD + 0.5/WARM + 0.5/HOT}, that do not match any numerical value.

2.2.1 Fuzzified crisp data

Each component of a crisp piece of data that passes through the mentioned fuzzification stage is converted into a probability distribution over the set of linguistic labels, i.e. a fuzzy random variable (frv). Let X and Y be two fuzzified measurements of crisp vectors, $X = (x_1, \ldots, x_n)$ and $Y = (y_1, \ldots, y_n)$. We propose that the distance between X and Y is the euclidean distance

$$d(X, Y) = \left(\sum_{i=1}^{n} d(x_i, y_i) \right)^{0.5}. \qquad (8)$$

The distance between each component depends, in turn, of the probabilities that each label has been assigned. Let $(p_1, \ldots, p_l)$ and $(q_1, \ldots, q_l)$ the probabilities assigned to these labels $(\widetilde{L}_1, \ldots, \widetilde{L}_l)$ in x_i and y_i. We propose that the distance between them is a fuzzy set: the distance between the expectations of both frv,

$$\widetilde{d}(x_i, y_i) = || \bigoplus_{i=1}^{l} (p_i - q_i)\, \widetilde{L}_i ||. \qquad (9)$$

For reducing the computational time, we have replaced eq. (9), by the approximate crisp distance

$$d(x_i, y_i) = | \sum_{i=1}^{l} (p_i - q_i) \operatorname{cog}(\widetilde{L}_i) |, \qquad (10)$$

where $\operatorname{cog}(\widetilde{L}_i)$ is the center of gravity of $\widetilde{L}_i$.

2.2.2 Missing values and vague data

The memberships of either a missing value or an imprecisely measured data can be understood as a family of probability distributions. We can determine a lower and an upper bound of the distances between these pieces of information as the interval

$$\begin{aligned} d(x_i, y_i) \quad = \quad & \\ & \{ | \textstyle\sum_{i=1}^{l} (p_i - q_i) \operatorname{cog}(L_i) | : \\ & p_i \in [p_{i*}, p_i^*],\ q_j \in [q_{j*}, q_j^*] \} \end{aligned} \qquad (11)$$

Note that, in this case, the situation is equivalent to the case studied in the preceding section, when the data was imprecisely measured and the imprecision was defined by means of crisp sets. Let us call r to the radius of the volume V in the preceding section. We can define $\hat{P}_*(C|x)$ and $\hat{P}^*(C|x)$ as the minimum and maximum of the set

$$\left\{ \frac{\sum_C a_i}{\sum a_i} : a_i \in \left\{ \begin{array}{ll} \{0\} & \min\{d(S_i, x)\} > r \\ \{1\} & \max\{d(S_i, x)\} < r \\ \{0, 1\} & \text{else} \end{array} \right. \right\} \qquad (12)$$

and use the rule in eq. (6) to obtain the set of classes that the object is assigned.

2.3 Measurement of the error rate of a classifier with imprecise data

For computing the error rate of the classifier over a dataset we want to count the number of misclassifications. However, since the output of the classifier is a set of classes, it is not always possible to decide whether the point is being correctly classified. Generally speaking, the error rate will also be a fuzzy set.

Let us assume that the output of the fuzzy classifier for the j-th element of the dataset is the vector $(\mu_{C_1}, \ldots, \mu_{C_M})$. Let q be the index of the modal point of this set, and let b the index of the second highest membership. According to our proposal in [4], this classification contributes to the total error as much as

$$E_j = \begin{cases} 1/0 + \mu_{C_b}/1 & \text{if } C_j = C_q \\ \mu_{C_j}/0 + 1/1 & \text{else} \end{cases} \quad (13)$$

i.e., the number of errors is the sum, with fuzzy arithmetic, of the values E_j.

3 Genetic Search of the best subset of features

In case the data is crisp, the approximate crisp distance in eq. (10) reduces to an standard k-NN, thus conventional GAs can still be used. Nonetheless, there is a difference: the k-NN in our approach depends on the fuzzy membership functions, thus two different input values with the same membership will have null distance in the new method. As a matter of fact, in [7][8], we had already shown that there exist datasets where the ranking of the input variables was dependent on the fuzzy definition of the antecedents of the rules. If we use the feature selection algorithm over the (fuzzy) discretized data, we could detect those cases where the partition of the inputs has removed relevant information, and decide not to use these variables. As a consecuence of this, the algorithm proposed here can be used for selecting the most informative subset of inputs, and it is valid either for vague or fuzzified continuous data.

On the other hand, when the data is interval-valued, fuzzy or crisp with missing values, the fitness function will be a fuzzy number, and we will need a specially crafted multicriteria genetic algorithm [9, 10] in order to solve the problem. We give a brief explanation of this algorithm in the remainder of this section.

3.1 Representation and Genetic Operators

We have used the same representation and operators proposed in [16]. The subsets have fixed cardinality, thus we use integer coding. The length of a chromosome is the number of features, and each allele represents one variable.

Two different crossover operators are used: partially complementary crossover operator [18] and two point crossover with repair operation (i.e., repeated features are replaced by a non-selected variable).

3.2 Fitness function

The quality of a given subset is given by the average error rate in the test set of the k-NN classifier. Five training-test partitions are used in this evaluation. In case the input data is crisp and there are not missing values, the fitness function is a crisp number. In other case, it is a fuzzy number. That value is the sum (by means of fuzzy arithmetic operators) of the costs E_j of each test data, as defined in eq. (13).

3.3 Generational scheme

As described in [9, 10], we have used a generational approach with the multiobjective NSGA-II replacement strategy, binary tournament selection based on rank and crowding distance, and a precedence operator that assumes a uniform prior. The nondominated sorting depends on the product of the so-obtained probabilities of precedence. Lastly, the crowding is based on the Hausdorff distance.

4 Numerical analysis

The algorithm described in previous section is evaluated and the results are discussed in this section. Thirteen different fuzzy rule learning algorithms have been considered, both heuristic and genetic algorithms-based. The heuristic classifiers are described in [23]: no weights (HEU1), same weight as the confidence (HEU2), differences between the confidences (HEU3, HEU4, HEU5), weights tuned by reward-punishment (REWP) and analytical learning (ANAL). The genetic classifiers are: Selection of rules (GENS), Michigan learning (MICH) –with population size 25 and 1000 generations–, Pittsburgh learning (PITT) –with population size 50, 25 rules each individual and 50 generations–, and Hybrid learning (HYBR) –same parameters than PITT, macromutation with probability 0.8– [23]. Lastly, two iterative rule learning algorithms are studied: Fuzzy Ababoost (ADAB) –25 rules of type I, fuzzy inference by sum of votes– and Fuzzy Logitboost (LOGI) –10 rules of type III, fuzzy inference by sum of votes– [7]. All the experiments have been repeated ten times for different permutations of the datasets (10cv experimental setup).

Because of space reasons, we limit ourselves to crisp data and study the effect of including information about the fuzzy partition in the feature selection algorithm. In Table 1 we have compared the results of the new algorithm FMIFS for five crisp datasets to those of the original MIFS algorithm, the RELIEF [24] and the evolutionary algorithm SSGA [16]. In all cases, a uniform partition of size 3 was used for all the variables, and 5 input variables were selected. The algorithm FSSGA was not different from the best one in XX cases, while FMIFS was the best choice in XX, SSGA in X, RELIEF in X and the crisp version of MIFS was the best in X. Observe that there are two problems were both FMIFS and FSSGA improve the results of the crisp feature selection. In the remaining problems, the use of a fuzzy method did not degrade the results, and FSSGA is not different than its crisp version. Therefore, we think that this algorithm is a good compromise. In future works we will include compared results of the performance of FSSGA and FMIFS over coarsely measured data and data with missing values.

It is worth pointing that that we do not claim that the use of fuzzified data always improves the performance of fuzzy classifiers. It depends on the linguistic partition. We claim that there exist cases where the linguistic partition has to be taken into account, and that those cases are not pathological: we have used uniform partitions, which are the most common in practical situations. In case the fuzzy partition is optimal, the gain is no longer relevant.

5 Concluding remarks

The preprocessing of databases with imprecise data is hardly found in the literature. In this paper we have proposed a wrapper-type evolutionary feature selection algorithm, with promising results. This algorithm can be applied to crisp and fuzzy problems, and we have shown that, even in the case that the data is crisp, there exist problems where we obtain a consistent improvement for the whole catalog of fuzzy systems that were tested. Intuitively, the method proposed here should be applied when the input partition has few elements and/or it has not been optimized, but further work is needed to characterize this family of problems.

The results in vague datasets have not been included because of space reasons (however, in vague datasets the benchmarks compare two algorithms of our own, thus the conclusions are less reproductible than the selection shown). Some works remains to be done (for example, determining the best number of features).

Acknowledgements

This work was funded by Spanish M. of Science and Technology and by FEDER funds, within the grant TIN2005-08036-C05-05.

References

[1] Siedlecki, W., Sklansky, J.: A note on genetic algorithms for large-scale feature selection. Pattern Recognition Letters 10 (1989) 335-347

[2] E. Cantú-Paz. Feature Subset Selection, Class Separability, and Genetic Algorithms. Genetic and Evolutionary Computation Conference (GECCO04). Lecture Notes in Computer Science 3102, Springer-Verlag 2004, Seattle (USA, 2004) 959-970

	HEU1	HEU2	HEU3	HEU4	HEU5	REWP	ANAL	GENS	MICH	PITT	HYBR	ADAB	LOGI	best
GERMAN - RELIEF	0.295	0.285	0.275	0.275	0.275	0.280	0.275	0.270	**0.295**	0.285	0.295	0.290	0.260	1
GERMAN - SSGA	0.265	**0.255**	**0.250**	**0.255**	**0.255**	**0.250**	0.260	0.255	**0.295**	**0.275**	**0.255**	**0.260**	0.255	9
GERMAN - MIFS	0.280	0.265	0.265	0.265	0.265	0.265	0.260	0.265	**0.295**	**0.275**	0.285	0.265	**0.250**	3
GERMAN - FMIFS	**0.255**	**0.255**	0.255	**0.255**	**0.255**	0.260	**0.245**	**0.250**	0.305	**0.275**	**0.255**	0.265	0.270	8
GERMAN - FSSGA														
ION - RELIEF	0.328	0.314	0.285	0.285	0.285	0.200	0.257	0.157	0.428	0.228	0.214	**0.114**	0.142	1
ION - SSGA	0.200	0.185	0.157	0.157	0.157	0.142	**0.157**	0.128	0.328	**0.114**	**0.114**	0.514	0.100	3
ION - MIFS	0.200	0.200	0.200	0.200	0.200	0.185	0.185	0.185	0.357	0.157	0.142	0.514	0.171	0
ION - FMIFS	**0.185**	**0.142**	**0.128**	**0.128**	**0.128**	**0.128**	0.171	**0.100**	**0.200**	**0.114**	0.128	0.514	**0.085**	10
ION - FSSGA														
PIMA - RELIEF	0.289	0.289	0.276	0.276	0.276	0.269	0.269	0.263	**0.355**	**0.230**	0.256	0.243	0.250	2
PIMA - SSGA	0.302	0.289	**0.263**	**0.263**	**0.263**	**0.263**	**0.263**	0.263	**0.355**	0.243	**0.243**	**0.217**	**0.217**	9
PIMA - MIFS	**0.276**	**0.276**	0.276	0.276	0.276	0.276	0.276	0.269	**0.355**	0.256	0.276	0.223	0.243	3
PIMA -FMIFS	0.302	0.289	**0.263**	**0.263**	**0.263**	**0.263**	**0.263**	**0.243**	**0.355**	0.250	0.276	**0.217**	**0.217**	9
PIMA - FSSGA														
SONAR - RELIEF	0.300	0.275	0.250	0.250	0.250	0.275	0.375	0.300	**0.300**	**0.275**	0.325	0.300	0.250	2
SONAR - SSGA	0.300	0.325	0.250	0.250	0.250	0.300	0.325	0.250	**0.300**	0.300	0.250	0.250	0.250	1
SONAR - MIFS	0.350	0.325	0.300	0.300	0.300	0.350	0.350	0.250	0.350	0.325	0.350	0.350	0.325	0
SONAR - FMIFS	**0.225**	**0.200**	**0.175**	**0.175**	**0.175**	**0.200**	**0.225**	**0.175**	**0.300**	**0.275**	**0.225**	**0.150**	**0.200**	13
SONAR - FSSGA														
WINE - RELIEF	0.500	0.411	0.235	0.205	0.176	0.088	0.235	**0.029**	0.647	0.205	**0.029**	0.058	0.058	2
WINE - SSGA	**0.176**	0.176	0.147	0.235	**0.147**	**0.058**	**0.088**	**0.147**	0.147	0.058	**0.029**	**0.000**	**0.029**	8
WINE - MIFS	0.323	0.323	0.264	0.205	0.176	0.117	0.235	0.176	0.617	0.058	0.176	0.058	0.058	0
WINE - FMIFS	**0.176**	**0.147**	**0.117**	**0.176**	**0.147**	**0.058**	0.147	0.117	0.176	**0.029**	0.088	0.058	0.058	7
WINE - FSSGA														

Table 1: 10 fold cross validation-based test error of different fuzzy rule-based classifiers after a feature selection was performed. The number of times each algorithm was the best is shown in the last column. FMIFS, SSGA, Relief and MIFS were the best 47, 30, 8 and 6 times, respectively. 5 input variables and 3 linguistic labels were used for each variable.

[3] P. Zhang, B. Verma, K. Kumar. Neural vs. statistical classifier in conjunction with genetic algorithm based feature selection. Pattern Recognition Letters (2007) In press

[4] L. Sanchez and I. Couso, Advocating the use of imprecisely observed data in genetic fuzzy systems, IEEE Transactions on Fuzzy Systems, vol. 15, pp. 551-562, 2007.

[5] Cost, S., Salzberg, S. A weighted nearest neighbor algorithm for learning with symbolic features. Machine Learning 10 (1), 57-78. 1993

[6] Li, Y., Wu, Z.-F. "Fuzzy feature selection based on min-max learning rule and extension matrix", Pattern Recognition 41, pp. 217–226. 2008.

[7] Sánchez, L., Suárez, M. R., Villar, J. R., Couso, I. Some Results about Mutual Information-based Feature Selection and Fuzzy Discretization of Vague Data, FUZZ-IEEE 2007, London. 2007.

[8] Sánchez, L., Suárez, M. R., Villar, J. R., Couso, I. Mutual Information-based Feature Selection and Partition Design in Fuzzy Rule-based Classifiers from Vague Data. Int. J. Approximate Reasoning, submitted.

[9] Sánchez, L.; Couso, I.; Casillas, J. "A Multiobjective Genetic Fuzzy System with Imprecise Probability Fitness for Vague Data". Proc. Int. Symp. on Evolving Fuzzy Systems, pp. 131-136, 2006.

[10] Sánchez, L., Couso, I., Casillas, J. "Modelling vague data with genetic fuzzy systems under a combination of crisp and imprecise criteria". Proc. 2007 IEEE MCDM, Honolulu, USA,. 2007

[11] Sánchez, L., Otero, J., Villar, J. R. "Learning fuzzy linguistic models from low quality data by genetic algorithms". Proc. FUZZ-IEEE 2007, London. 2007.

[12] Uncu, O., Türksen, I.B. "A novel feature selection approach: Combining feature wrappers and filters", Information Sciences, 177 (2), pp. 449-466. 2007.

[13] Sun, H.-J., Sun, M., Mei, Z. "Feature selection via fuzzy clustering". Proc. 2006 Int. Conf. on Machine Learning and Cybernetics 2006, art. no. 4028283, pp. 1400-1405. 2006.

[14] Zhang, Y., Wu, X.-B., Xiang, Z.-R., Hu, W.-L. "Design of high-dimensional fuzzy classification systems based on multi-objective, evolutionary algorithm", Journal of System Simulation, 19 (1), pp. 210-215. 2007.

[15] Xiong, N., Funk, P. "Construction of fuzzy knowledge bases incorporating feature selection", Soft Computing, 10 (9), pp. 796-804. 2006.

[16] Casillas, J., Cordón, O., del Jesus, M. J., Herrera, F. "Genetic Feature Selection in a Fuzzy Rule-Based Classification System Learning Process", Information Sciences 136 (2001) 135-157.

[17] H. Liu and R. Setiono, "A probabilistic approach to feature selection - a filter solution", In Machine Learning, Proc. of the 13th International Conference, Bari, Italy, pp. 319-327.

[18] W.Liu, M. Wang, Y. Zhong, Selecting features with genetic algorithm in handwritten digits recognition. Proc Int. IEEE Conf. Evol. Comp. (ICEC'95), 1, 1995, 396-399

[19] J.M. Keller, M.R. Gray, J.A. Givens Jr., "A fuzzy k -nearest neighbor algorithm", IEEE Trans. Systems Man Cybernet. vol 15, 4, pp. 580–585. 1985.

[20] W.-L. Huang, H.-M. Chen, S.-F. Hwang and S.-Y. Ho, "Accurate prediction of enzyme subfamily class using an adaptive fuzzy k-nearest neighbor method", BioSystems, vol. 90, 2007, 405–413

[21] K. Toduka and Y. Endo, "Fuzzy K-Nearest Neighbor and its Application to Recognize of the Driving Environment", In the 2006 IEEE International Conference on Fuzzy Systems, Vancouver, BC, Canada, pp. 751–756

[22] M. Sarkar, "Fuzzy-rough nearest neighbor algorithms in classification", Fuzzy Sets and Systems, vol. 158, 2007, 2134–2152

[23] H. Ishibuchi, T., Nakashima and M., Nii. 'Classification and Modeling with Linguistic Information Granules. Springer. 2004.

[24] Kira, K. and Rendell, L. A practical approach to feature selection. In: Sleeman and P. Edwards (Eds) Proceedings of the Ninth International Conference on Machine Learning (ICML-92), Morgan Kaufmann, 249-256. 1992.