Proceedings of the 7th WSEAS International Conference on Fuzzy Systems, Cavtat, Croatia, June 12-14, 2006 (pp33-38)

The Hybrid Genetic Fuzzy C-Means: a Reasoned Implementation

ALESSANDRO G. DI NUOVO, VINCENZO CATANIA, MAURIZIO PALESI
Dipartimento di Ingegneria Informatica e delle Telecomunicazioni, Università degli Studi di Catania
Viale Andrea Doria 6, 95125 Catania, ITALY

Abstract: - In this paper we present an hybrid approach which integrate Fuzzy C-Means (FCM) algorithms and Genetic Algorithms (GAs) to design an optimal classifier for the specific classification problem. This integration allows automatic generation of an classifier system, with an optimized subset of features, from a database of examples. The generated classifier strongly outperform the classic FCM algorithm. A reasoned implementation of the hybrid algorithm, we called GFCM, is given along with a comparative study and performance evaluation results on several public benchmark databases. Results obtained show the efficiency of GFCM algorithm.

Key-Words: - Fuzzy C-Means, Genetic Optimization of Fuzzy Classifier, Features Selection and Weighting, Pattern Classification.

1 Introduction

The most widely used clustering algorithm implementing the fuzzy philosophy is Fuzzy C-Means (FCM), initially developed by Dunn and later generalized by Bezdek, who proposed a generalization by means of a family of objective functions [1]. Despite this algorithm proved to be less accurate than others, its fuzzy nature and the ease of implementation made it very attractive for a lot of researchers, that proposed various improvements and applications (for examples refer to [2]). Usually FCM is applied to unsupervised clustering problems. In this paper, however, we show how it can be applied successfully to supervised classification problems. Thanks to the integration with a Genetic Algorithm (GA) [3], an hybrid Genetic Fuzzy C-Means (GFCM) is built up to concurrent solve the supervised classification and the feature selection problems.

Several works have been proposed in the literature which make use of Evolutionary Algorithms (EAs) for fuzzy clustering, some of them are devoted to improve the performance of FCM-type algorithms [4] using the GA to optimize parameters of these algorithms, others are designed to create directly a fuzzy partition of data.

The fuzzy systems, which use GA to learn their structure from examples and to improve their performances, are called Genetic Fuzzy Systems (GFSs) [5]. The use of GAs to optimize the parameters of an FCM-type algorithm generates two different kinds of GFSs. Prototype-based algorithms encode the fuzzy cluster prototypes and evolve them by means of a GA guided by any centroid-type objective function [6], while fuzzy partition-based algorithms encode, and evolve, the fuzzy membership matrix [7].

A second possibility is to use the GA to define the distance norm of an FCM-type algorithm. The system considers an adaptive distance function and employs a GA to learn its parameters to obtain an optimal behavior of the FCM-type algorithm [8].

A third group of genetic approaches are based on directly solving the fuzzy clustering problem without interaction with any FCM-type algorithm. These techniques, which are a recent trend in cluster analysis, have shown the potential to achieve high partitioning accuracy results. Previous approach employed Evolutionary Strategies [4], Evolutionary Programming [9], and recently Particle Swarm Optimization [10] and Simulated Annealing [11]. Details about other types of clustering are to be found in [12].

Many search algorithms have been used for feature selection [13]. Among these, EAs have proven to be an effective computational method, especially in situations where the search space is uncharacterized (mathematically), not fully understood, or/and highly dimensional.

One particular application of these methods not only selects features but also assigns them weights according to their importance for the analysis to be performed [14].

Feature selection techniques do not normally offer the possibility of classifying the sets they analyze: they are, in fact, proposed often as filter techniques. There are, however, certain exceptions such as C4.5 [15], which belongs to the supervised machine learning category. More recent work that integrates feature selection with classificatory analysis has been presented in [16].

The rest of the paper is organized as follows. Section 2 describes the solutions adopted to integrate the two algorithms and to improve GFCM performances. Section 3 presents numerical results on several benchmark database and comparisons with some similar approaches. Finally Section 4 gives our conclusions.

2 The Genetic Fuzzy C-Means Algorithm

In this section we present the GFCM approach with detailed demonstrations of the solutions adopted in our implementation. In our experiments we use five databases (Iris, Wine, Diabetes, New Thyroid, Sonar) which are available for free download at [17], along with the relative information.

Fig. 1 explain how the GFCM works. Essentially it can be represented as a feedback controlled system, where the FCM is the system to be controlled and the GA is the controller, which tune the FCM parameters evaluating its output (i.e. classification results).

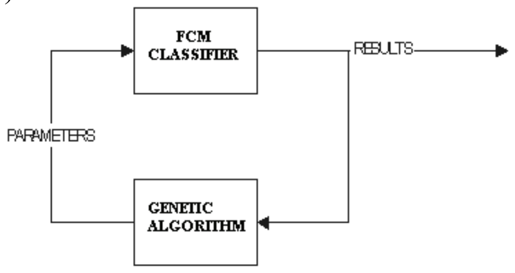

Fig. 1. Systemic representation of the proposed approach.

The FCM version we implemented proposes to minimize the following objective function:

$$J(U,C;X)=\sum_{j=1}^{K}\sum_{i=1}^{N}u_{ij}^{m}\cdot d(\mathbf{x}_i,\mathbf{c}_j) \qquad (1)$$

where $\mathbf{c}_j$ is the prototype of the j-th cluster and $d(\bullet,\bullet)$ is a distance metric appropriately chosen from the pattern space, $\mathbf{x}_i$ is the i-th pattern, u_{ij} is the degree of truth of the i-th pattern in the j-th cluster, raised to the "fuzzyfier" m. K and N are respectively the number of clusters and the number of patterns. m is a parameter on which the degree of fuzzyfication depends: as its value increases, so does the degree of uncertainty, until it settles at $u_{ij} = 1/K\ \forall i,j$, whereas when it gets close to 1 the result is the partitioning typical of "hard" algorithms.

The accuracy by which the algorithm classifies patterns depends heavily on the value of m. Normally it is made to vary between 1.5 and 2, but sometimes better classification is achieved with higher or lower values. The optimal choice is linked to the data set being investigated and thus varies according to the application involved. There are no techniques which allow an effective value to be chosen a priori. In our implementation we use the genetic algorithm to estimate the optimal value for m.

The FCM algorithm is summarized in the following steps :

1. Given the N patterns assume a predefined number of clusters K, where $K\in[2,N[$.

2. At step p=0, initialize the matrix $U^{(0)} = [u_{ij}]$, respecting the constraint given by

$$\sum_{j=1}^{K}u_{ij}=1 \quad \forall i \qquad (2)$$

3. At step p>0, calculate the centroids $C^{(p)}$, i.e. the prototype vectors $\mathbf{c}_j$, starting from $U^{(p-1)}$:

$$\mathbf{c}_j=\frac{\sum_{i=1}^{N}u_{ij}\cdot\mathbf{x}_i}{\sum_{i=1}^{N}u_{ij}} \qquad (3)$$

4. Update the matrix $U^{(p-1)}$, obtaining $U^{(p)}$:

$$u_{ij}=\frac{1}{\sum_{k=1}^{K}\left[\frac{d(\mathbf{x}_i,\mathbf{c}_j)}{d(\mathbf{x}_i,\mathbf{c}_k)}\right]^{2/(m-1)}} \qquad (4)$$

5. Verify whether the STOP criterion is satisfied; if it is not, return to point 3.

The STOP criterion normally chosen is $\|U^{(p)} - U^{(p-1)}\| < \varepsilon$, with $\varepsilon\geq 0$.In order to avoid long calculation time it is preferable to choose a certain number of iterations as the STOP criterion; in this case the first condition is also applied and the algorithm is stopped when one of the two is met.

2.1 Initialization of the centroids

Initialization of the centroids (i.e. of the prototype vectors $\mathbf{c}_j$) is still an open issue. Many examples have been published showing how solutions obtained by a c-means algorithm strongly depend on the starting condition [2]. There is no general agreement about a good initialization scheme.

An attempt to address this problem was presented in [18]. The basic idea is to insert the initial values of the centroids among the parameters to be optimized by a GA. Using this approach authors observed an increase in computation time of two orders of magnitude as compared to normal execution. In normal conditions it is therefore preferable to execute the algorithm several times, starting from different initial values, which gives similar, if not identical, results.

In this work we present GFCM, as a learning classification algorithm, which is able to implement an optimal classifier mining data sets whose patterns are already been classified. For this reason we chose to initialize the centroids with the average values of the predefined groups. By using this kind of initialization, we found that the solutions are equivalent to those obtained by the GA approach proposed in [18], with a considerable saving in computation time.

Table 1. Objective function (Obj_fcn) value and classification error (%), a comparative study of initialization methods.

Data Set	Best of 100 repetitions with random init		GA approach		Average Init	
	Obj_fcn	Error%	Obj_fcn	Error%	Obj_fcn	Error%
Iris	60.51	10.7	60.51	10.7	60.51	10.7
Wine	1796082.79	30.9	1796078.66	30.3	1796164.25	30.3
Diabetes	3986862.80	34.8	3986861.92	34.8	3996643.25	35.0
New Thyroid	11933.92	33.2	11781.99	32.8	11779.35	32.8
Sonar	180.36	45.6	180.31	45.2	180.41	45.2

A comparative study between the three initialization methods (random, GA, average) is presented in Table 1, which shows a substantial equivalence between them all.
The results for GA were obtained in two distinct phases, in the first one the goal was to minimize the objective function (1), in the second one the goal was to minimize the classification error.

2.2 Distance measure

Different distance metrics are used by different authors. At any rate there is not any clear indication about which of them provide the best results. The data themselves have the last say about which distance could provide best results.
In our work we chose a variable Minkowski metric with a weights matrix, which makes the algorithm extremely flexible [8] and allows it to adapt to sets of any shape [19]. This is also beneficial for centroid initialization: as the space can be varied at will, the genetic algorithm will be able to find the parameters that will guarantee optimal convergence even though the initialization is not equally optimal.
The metric chosen was:

$$d(\mathbf{x},\mathbf{y}) = \sqrt[p]{\sum_{k=1}^{D} w_k^p (x_k - y_k)^p} \qquad (5)$$

where D is the size of the space of features and w_k is the weight assigned to the k-th feature, which is inserted as a parameter to be estimated by the GA. The dimensionality p will also be estimated by the GA.

2.3 Normalization of the data

As for the distance metric, the right choose of the normalization method is still an open issue. There is not any method that is optimal for every classification problem. For this reason in this subsection we analyze the normalization impact on FCM classification performance.
In order to chose the normalization method that best fit our GFCM algorithm we compared different normalization methods in terms of classification performance for different datasets. In tests we used the Euclidean metric (i.e. p=2 in equation (5)) as distance measure and the initialization method was the one that uses average values of the predefined groups.
The normalization methods we compared are:

1. Classical normalization, $X_{norm} = (X - min)/(max-min)$.
2. Rescaling normalization, the division by the maximum number, $X_{norm} = X / max$.
3. Non Linear normalization, $X_{norm} = (X/max)^2$.

Table 2. Normalization methods comparison: classification error (%) of Euclidean FCM with m = 2.

Data Set	No Norm Error%	Classical Norm error %	Rescaling Norm error %	Non linear Norm Error %
Iris	10.7	11.3	4.0	8.0
Wine	30.3	5.0	7.9	13.5
Diabetes	35.0	28.9	30.2	27.0
New Thyroid	32.8	9.8	8.4	31.6
Sonar	45.2	40.9	39.4	43.7

Table 2 shows clearly that data normalization is very useful to improve classification quality, in fact all three normalization methods considered improve the FCM performances. However there is no method which dominate the others for all the data sets. The rescaling normalization method can be seen as a particular case of a more general non linear normalization method, where $X_{norm} = (X/max)^\alpha$, with $\alpha > 0$.

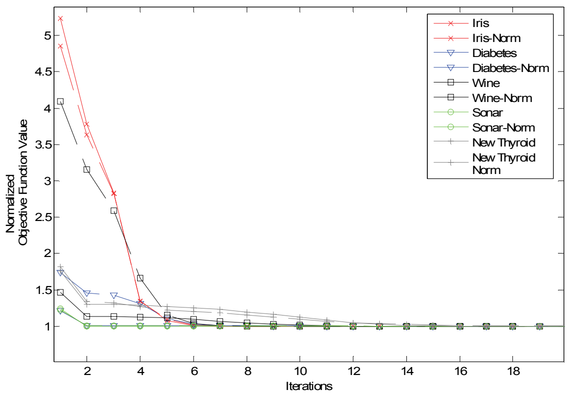

Figure 2. Objective function value (normalized) for varying number of iterations. Normalization method used was the rescaling one.

Table 2 shows that the general non linear normalization method leads often to the best accuracy, so we decided to use this one in our implementation and let the GA search for the best α. These tests were also useful to set the stop criterion for FCM. Figure 2 shows the objective function values for varying numbers of iterations. As can be seen no appreciable improvements in objective function are found after the 15th generation. So we set at 15 the maximum number of iteration beyond whom the FCM stops even if the error criterion was not satisfied.

From Figure 2 it can be seen that there are no appreciable differences in number of iteration required by FCM to converge with and without rescaling norm.

2.4 GA operators

A GA operates on a population of individuals called "chromosomes", each representing a candidate solution to a given problem. An iteration of the algorithm, known as a generation, causes the current population to evolve into a new one with the aim of finding increasingly better candidates. The evolution of a GA derives from selection and reproduction (copy, crossover, and mutation) operators.

Selection is used to determine which individuals could be used for the following Reproduction operators. According to the roulette rule, the algorithm will choose the individuals in relation to their fitness: the "fitter" a chromosome, the greater its probability of being chosen. The individuals selected for reproduction are called "parents".

The first Reproduction operator is Copy. It only chooses a parent and copies it in the new generation. This technique prevents the elimination of the best individuals which could cause a regression to worse candidates. Crossover is the main reproduction operator and it exchanges portions of the chromosome of two parents to create two new individuals called "offspring". The third reproduction operator is Mutation, which allows new parts of a chromosome to be inserted into a parent. It is useful to get out of local minimum or maximum values. These operators are selected in a probabilistic way; each operator probability is defined before the algorithm starts.

In our specific case, the mutation operator randomly modifies the value of a parameter chosen at random. The crossover between two configurations exchanges the value of two parameters chosen at random. Mutation probability used in our real database tests was 0.3, crossover probability was 0.6.

2.5 The chromosome

The chromosome of the GA is defined with as many genes as there are free parameters and each gene will be coded according to the set of values it can take. In our case study, both the parameters of FCM and feature weights are mapped on a chromosome whose genes are gray binary coded with 8 bits precision.

The chromosome genes are: the normalization parameter α, the FCM fuzzyfier *m*, distance measure dimensionality *p*, and the *D* feature weights.

Table 3 summarizes the parameters search space.

Table 3. Parameters search space.

Parameter	Parameter Space	Step	Notes
α	[0.1,5.0]	0.1	
m	[1.1,5.0]	0.1	
p	[1.0,5.0]	0.1	
$w_1 \dots w_D$	[0.0,1.0]	0.1	We have forced the probability to be 0.0 to 50% in order to have the same chances either to select or to erase a feature.

2.6 Fitness Function

The definition of the fitness function plays a crucial role in the design of the GFCM. Its structure was chosen in order to achieve the highest classification accuracy. In fact, in GFCM design, the feature selection was a secondary characteristic, which is, on the other hand, useful to eliminate redundant and noisy features, to improve the classification accuracy.

The classification accuracy can be evaluated as the amount of patterns correctly classified, so we defined the vector **z**, where the elements z_i are binary variable whose value is 1 if the predicted group is equal to the real one and 0 otherwise. The dimensionality of vector **z** is *M*, which is the number of patterns with a reference classification.

The objective of the GA is to maximize the fitness function (*F*), which is defined as follows:

$$F(\mathbf{z},\mu,\chi)=\sum_{i=1}^{M} z_i-\frac{\mu}{\chi} \qquad (6)$$

where μ is the number of features selected (i.e. the number of weights $w_i > 0$) and χ is a damping coefficient. The coefficient χ was introduced in order to strengthen or to weaken the selection of the features. However, thanks to the feature weights assigned by the GA, it is possible to made a further selection among the subset indicated by GFCM. Lower is the weight associated to a feature, lower is its significance, lower is the accuracy loss if it is deselected.

3 Performance and Comparative Studies

In the comparative tests described in this section the stop criterion used for the FCM algorithm was the achievement of a maximum variation lower than 0.01 or 15 iterations. For the GA, the maximum number of generation was set to 200. The GA population size was set to 2×*D*. The χ parameter of (6) was set as equal to *D*. Weights range was [0,1]. These values were set following an extended tuning phase in such a way as to be effective for all the data sets considered.

Table 4. GFCM classification performance improvements against simple Euclidean FCM (on whole sets).

Data Set	FCM Error%	FCM with Rescaling Norm error %	GFCM Error%	Improvement Factor X
Iris	10.7	4.0	2.0	2.00
Wine	30.3	7.9	1.1	7.18
Diabetes	35	30.2	22.5	1.34
New Thyroid	32.8	8.4	3.3	2.55
Sonar	45.2	39.4	18.9	2.08

First of all we would remark that GFCM dramatically improve classification performances in comparison with the simple Euclidean FCM, as show in Table 4 in which the improvement factor varies from 1.34X to 7.18X.

In this subsection we compare the GFCM with three feature selection and classification methods:

- C4.5 [15], the most famous algorithm that belongs to the class of decision tree methods.
- Ahmad&Dey [16], a recent feature selection algorithm based on conditional probabilities and k-nn classifier.
- IMOEA [20], that uses an intelligent multi-objective evolutionary algorithm to design k-nn classifiers.

The first two algorithms, like GFCM, gives information about significance of features.

Table 5 and Table 6 report classification error and feature selection performance for the different algorithms applied on several datasets [17, 21]. These datasets refeer to different application domains such as credit dataset, image dataset, dataset with costs, others. We chose at least one of them for each application domain (Australian Credit, Vehicle, Diabetes, Heart and DNA). For detailed description please refer to [17, 21]. As regards DNA, although it comprises 3186 elements with 180 features, we only used the 60 most significant features as suggested by the authors and as was done for the other algorithms.

In Table 6 the "percentage (%) of selected features" column shows the number of features indicated by GFCM to obtain the maximum accuracy.

To prove the effectiveness of GFCM on unseen patterns we applied the 10 fold cross-validation technique for our comparative studies, this practice is also useful to counterbalance the stochastic nature of GA.

As can be seen from Table 5, GFCM technique guarantees an accuracy higher or at worst similar than the one obtained with C4.5, which does not use feature significance. GFCM and Ahmad&Dey approach obtain comparable results both in classification accuracy and in feature selection (Table 6).

More in deep GFCM is almost slightly better than Ahmad&Dey approach in classification accuracy, meanwhile Ahmad&Dey approach select a less features than GFCM. These results are inline with our expectation because GFCM has been designed to favor accuracy than feature selection.

Table 5. Comparative study between GFCM and C4.5, Ahmad & Dey, Discriminant Analysis technique: classification error on test set (10 fold cross-validation).

Data Set	Number of groups	Number of patterns	C4.5 error %	Ahmad & Dey error %	GFCM error %
Iris	3	150	6.7	3.7	2.7
Aus-Credit	2	690	15.5	14.4	12.6
Vehicle	4	846	33.3	38.0	44.8
Diabetes	2	768	27.0	20.7	22.7
Wine	3	178	1.9	5.8	2.2
Heart	2	270	30.1	13.0	12.6
DNA	4	3186	7.6	6.5	8.8

Table 6. Comparative study between GFCM and Ahmad & Dey technique: feature selection performance.

Data Set	Number of features	Ahmad&Dey % of selected features	GFCM % of selected features
Iris	4	50.0	50.0
Aus-Credit	14	14.3	57.0
Vehicle	18	38.9	37.9
Diabetes	8	37.5	50.0
Wine	13	69.2	76.9
Heart	13	76.9	49.2
DNA	60	30.0	40.0

The poor performances of GFCM on Vehicle database are due to the high overlap between the classes and the presence of noise, which cause the FCM fail the classification.

Table 7. Feature selection and test set error rate comparison between IMOEA and GFCM.

Data Set	Number of patterns	Number of features	IMOEA error %	GFCM error %	IMOEA % of selected features	GFCM % of selected features
Iris	150	4	6.0	2.7	25.0	50.0
Wine	178	10	7.0	1.1	7.7	76.9
New Thyroid	215	5	5.4	3.8	20.0	60.0
Sonar	208	60	20.0	19.1	2.5	70.0

Table 7 and Table 8 show a comparison with IMOEA. indicate for GFCM the results after a pruning of features, based on their significance, in order to obtain an accuracy comparable with that provided by the IMOEA.

Table 7 highlights the different concepts of the two techniques: although they are both based on evolutionary algorithms, they exploit them in different

ways. Primary goal of GFCM is classification accuracy, bearing in mind the need to provide results that are as consistent as possible, and secondly to select the features, whereas IMOEA reduces the number of features and data as much as possible, sacrificing a certain amount of accuracy.
Results reported in Table 8 have been obtained by selecting the features in the order indicated by the algorithm until a comparable accuracy was reached. The results aim to obtain the greatest selection possible so as to compare better the two algorithms. Once again we notice that an higher number of feature are selected by GFCM.

Table 8**.** Feature selection and relative error rate comparison between IMOEA and GFCM with a further selection based on weights associated to features.

Data Set	IMOEA error %	GFCM after further feature selection error %	IMOEA % of selected features	GFCM after further feature selection % of selected features
Iris	6.0	4.0	25.0	25.0
Wine	7.0	7.3	7.7	26.8
New Thyroid	5.4	6.0	20.0	40.0
Sonar	20.0	20.7	2.5	36.2

4 Conclusion

In this paper we have presented a Genetic Fuzzy System, called Genetic Fuzzy C-Means (GFCM), which proved to be a powerful extension of the famous Fuzzy C-Means algorithm. Essentially the GFCM algorithm is a learning algorithm which can mine databases to build an optimized and efficient classifier. In our empirical tests on several well known databases the classifiers build by GFCM proved to be drastically better than classical FCM classifier, obtaining an improvement up to 7 times in classification accuracy. In addition GFCM is also up to 2 times faster than FCM thanks to the feature selection. The technique has a number of qualities, despite the simplicity of the two integrated algorithms. It provides three different possibilities of analysis in a single algorithm: (i) Data classification; (ii) the assignment of weights to each feature, from which it is possible to obtain the significance of each feature; (iii) Feature selection on the basis of the weights assigned. An important feature, as we have seen in comparisons with similar techniques, is that the integrated algorithm achieves an excellent accuracy in both classification and feature selection. Along with a minimum number of parameters to be set, this considerably simplifies the work of researchers, who do not need to apply (and know) different algorithms, and it thus allows it to be applied in real everyday situations.

References:

[1] J. C. Bezdek, *Pattern Recognition with fuzzy objective function algorithms*. New York: Plenum Press, 1981.
[2] J. C. Bezdek, J. Keller, R. Krishnapuram, and N. R. Pal, *Fuzzy Models and algorithms for pattern recognition and image processing*: Springer, 1999.
[3] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*: Addison-Wesley, 1989.
[4] R. Babuska, *Fuzzy Modeling for Control*. Dordrecht: Kluwer Academic Press, 1998.
[5] O. Cordón,F. Herrera,F. Hoffmann, and L. Magdalena, *GENETIC FUZZY SYSTEMS Evolutionary Tuning and Learning of Fuzzy Knowledge Bases*, World Scientific,2001.
[6] L. O. Hall, J. C. Bezdek, S. Boggavarapu, and A. Bensaid, "Genetic fuzzy clustering," presented at NAFIPS'94, San Antonio,Texas, USA, 1994.
[7] T. Van Le, "Evolutionary fuzzy clustering," presented at IEEE Conf. on Evolutionary Computation (ICEC'95), Perth, Australia, 1995.
[8] B. Yuan, G. J. Klir, and J. F. Swan-Stone, "Evolutionary fuzzy c-means clustering algorithm," presented at FUZZ-IEEE '95, 1995.
[9] D. B. Fogel and P. K. Simpson, "Evolving fuzzy clusters," presented at International Conference on Neural Networks, 1993.
[10] D. W. van der Merwe and A. P. Engelbrecht, "Data clustering using particle swarm optimization," presented at Congress on Evolutionary Computation,Australia,2003.
[11] S. Bandyopadhyay, "Simulated Annealing Using Reversible Jump Markov Chain Monte Carlo Algorithm for Fuzzy Clustering'," *IEEE Transactions on Knowledge and Data Engineering* vol. 17, pp. 479-490, 2005.
[12] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data Clustering: A review," *ACM Computing Surveys*,vol.31, 1999.
[13] A. Jain and D. Zongker, "Feature Selection: Evaluation, Application, and Small Sample Performance," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, pp. 153-158, 1997.
[14] F. Hussein, R. K. Ward, and N. N. Kharma, "Genetic Algorithms for Feature Selection and Weighting, A Review and Study," presented at ICDAR 2001, 2001.
[15] J. R. Quinlan, *C4.5: Programs for Machine Learning*: Morgan Kauffman, 1993.
[16] A. Ahmad and L. Dey, "A feature selection technique for classificatory analysis," *Pattern Recognition Letters*, vol. 26, pp. 43-56, 2005.
[17]http://www.ics.uci.edu/~mlearn/MLRepository.html.
[18] L. O. Hall, I. B. Ozyurt, and J. C. Bezdek,"Clustering with a genetically optimized approach," *IEEE Trans. on Evolutionary Computation*,vol. 3,pp.103-112,1999.
[19] L. Bobrowski and J. C. Bezdek, "C-means clustering with the l_1 and $l_{¥}$ norms," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 21, pp. 545-554, 1991.
[20] J.-H. Chen, H.-M. Chen, and H. S.-Y., "Design of Nearest Neighbor Classifiers Using an Intelligent Multi-Objective Evolutionary Algorithm," at PRICAI,2004.
[21]http://www.niaad.liacc.up.pt/old/statlog/datasets.html.