Multi-Objective Evolutionary Fuzzy Clustering for High-Dimensional Problems

Alessandro G. Di Nuovo, Maurizio Palesi, Vincenzo Catania

***Abstract*— This paper deals with the application of unsupervised fuzzy clustering to high dimensional data. Two problems are addressed: groups (clusters) number discovery and feature selection without performance losses. In particular we analyze the potential of a Genetic Fuzzy System, that is the integration of a multi-objective evolutionary algorithm with a fuzzy clustering algorithm. The main characteristic of the integrated approach is the ability to handle the two problems at the same time, suggesting a Pareto set of trade-off solutions which could have a better chance of matching the real needs. We exhibit the high quality clustering and features selection results by applying our approach to a real-world data set.**

I. Introduction

Clustering methods provide a useful tool to explore data. They aim at discovering groups (clusters) in a set of patterns such that the similarity among patterns in the same group is higher than that of patterns belonging to different clusters. Clustering algorithms are traditionally divided into three main groups: Hierarchical, Partitioning, and Distance-Based. Details of these and all other types of clustering are to be found in [1].Hierarchical clustering algorithms construct hierarchies of clusters in a top-down (agglomerative) or bottom-up (divisive) fashion. Hierarchical clustering algorithms have proved to yield high-quality results especially for applications involving clustering text collections. Nonetheless, their high computational requirements usually limit their use in some real-life applications, where the number of samples and their dimension is typically high (the cost is the square of the number of samples). Partitioning clustering algorithms start from an initial clustering (that can be randomly formed) and create partitioning by iteratively adjusting the clusters based on the distance between the data points and a representative member of the cluster. The most commonly used partitioning clustering algorithm is k-means. This algorithm initializes K centers and iteratively assigns each data point to the cluster whose centroid minimizes the Euclidean distance from the data point. Algorithms of this type can give good clustering results at a low cost, since their running time is proportional to KN, where N is the number of patterns present in the dataset. However, their results rely heavily on their initialization and they can converge to arbitrary local optima. Distance based clustering algorithms create a partitioning by considering neighbors of data points. Clusters are considered as high density neighborhoods of data points. Although the density parameter is critical for the successful application of distance based algorithms, recently proposed heuristics appear to yield high quality results. A recent trend in cluster analysis is evolutionary clustering, based on well-known evolutionary algorithms (EAs), which have shown the potential to achieve high partitioning accuracy. In fact, EAs have proved to be widely applicable with success by choosing the right criterion to optimize, whereas classical clustering approaches are often good only for certain problems. Let us consider, for example, a high-dimensionality dataset, where dimensionality is equal to the number of features a pattern can have. Often a larger number of features is needed to describe the characteristics and classify the pattern better. Many clustering approaches suffer from being applied in high-dimensional spaces. For instance, in K-means (KM) clustering, based on iteratively computing distances and cluster averages, increasing the data space dimensionality may introduce a large number of sub-optimal solutions (local minima). A further problem is related to distances in high space dimensionality. Defining clusters on the basis of distance requires that distances can be estimated. However, there are results [2] stating that when the space dimensionality is high the distance from a point to its farthest neighbor and to its nearest neighbor tend to become equal. Therefore the evaluation of distances, and the concept of ”nearest neighbor” itself, become less and less meaningful with growing dimensions. To tackle this problem a number of algorithms able to reduce the dimensionality of the space before or during the clustering process have been presented [3]. Evolutionary clustering also suffers from these problems, but it has the ability to solve them via a dimensionality reduction (i.e. feature selection) directly during its evolution [4], [5]. This practice often leads to better solutions, but, if abused, it could lead to less accurate results. In this paper we want to avoid this drawback thanks to the use of multi-objective optimization, with the two aims of reducing dimensionality and preserving an acceptable level of clustering accuracy. In many practical problems, the actual number of clusters is not known a priori, thus in our proposal we take this into account where neither reference classification nor the number of clusters are previously known. For this reason, we chose to assess the performance of the partitioning by means of a cluster validity index. In this paper we use the Xie-Beni (XB) [6] cluster validity index as the underlying optimizing criterion since it has been shown to outperform other indexes in several experiments [7]. The rest of the paper is organized as follows. In Section II, we recall some related works on evolutionary clustering and feature selection. Next, in Sections III and IV we briefly describe the fuzzy clustering, the XB index and the evolutionary algorithms which are the pillars of our

The authors are with the Dipartimento di Ingegneria Informatica e della Telecomunicazioni, Università di Catania, Viale A. Doria 6, 95125 Catania, Italy (email:{adinuovo,mpalesi,vcatania}@diit.unict.it).

approach. The proposed approach is described in Section V. In Section VI we apply the proposed approach to a real case study. Finally the paper ends with concluding remarks in Section VII.

II. Previous Work

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 [8] using the GA to optimize parameters of these algorithms, others are designed to directly create 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) [9]. The use of GAs to optimize the parameters of a FCM-type algorithm generates two different kinds of GFSs. A first group of genetic approaches are based on directly solving the fuzzy clustering problem without interaction with any FCM-type algorithm. These techniques have shown the potential to achieve high partitioning accuracy results [10], [11]. Prototype-based algorithms encode the fuzzy cluster prototypes and evolve them by means of a GA guided by any centroid-type objective function [12], while fuzzy partition-based algorithms encode, and evolve, the fuzzy membership matrix [13]. 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 [14]. Good results were also obtained through these hybrid approaches with classical clustering algorithms, especially the ones which integrate clustering and evolutionary algorithms to exploit the flexibility and adaptability of the EA together with the scalability and accuracy of classification algorithms. As said before, the increasingly high-dimensional data sets from many application domains have posed unprecedented challenges to clustering techniques, which are a fundamental step in the process of mining knowledge from data. To solve this problem, reducing the dimensionality of the space, the feature selection have become the focus of much research in areas of application improving the classification performances of the algorithms, providing both faster and more cost-effective predictors, as well as a better understanding of the underlying generation process. The aim of feature selection is to reduce the dimensionality of the problem, by eliminating irrelevant and redundant features, while simultaneously maintaining or enhancing classification accuracy. Many search algorithms have been used for feature selection for classification and clustering [15], [16]. 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. There are two kinds of feature selection algorithms [17], [3]:

- Filter feature selection algorithms, which remove the irrelevant characteristics without using a learning algorithm. They are efficient processes but the feature subsets obtained may not be the best ones for a specific learning process.
- Wrapper feature selection algorithms. This kind of feature selection algorithm selects feature subsets using the precision of a classification algorithm to evaluate each candidate subset. Their problem is inefficiency, since they have to execute the classification algorithm for each evaluation.

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 [18]. In particular in [19] authors show that an appropriate assignment of feature-weight can improve the performance of fuzzy c-means clustering. The weight assignment is given by learning according to the gradient descent technique. All the methods presented give just a single solution, which often it is not the best trade-off solution for the specific problem addressed. In facts several of the results presented in [20] show that some of the feature sets identified for different feature cardinalities are closely related. Given this structure in the decision space, the identification of all solutions in a single run should be more efficient than using individual runs of a single-objective optimization method. Therefore a more effective approach may be the consideration of clustering as a multi-objective optimization problem, as suggested by [21]. Usually no single best solution for this optimization task exists, but instead, the framework of Pareto optimality is embraced, where the algorithm gives a set of trade-off solutions, called Pareto set, among which it is possible to choose the one that is suitable for a specific work. In situations where the best solution corresponds to a trade-off between the different objectives only the multi-objective clustering algorithm will be able to find it. In literature there are some recent works about crisp multi-objective clustering. In [22] authors presented a novel evolutionary multi-objective local selection algorithm for unsupervised feature selection, called *ELSA*, to search for possible combination of features and numbers of clusters, with the guidance of two representative clustering algorithms. Morita *et al.* [23] make use of a multi-objective genetic algorithm where the minimization of the number of features and a validity index that measures the quality of clusters have been used to guide the search towards the more discriminant features and the best number of clusters. Recently Handl *et al.* in [24] presented *MOCK*, a multi-objective clustering algorithm with automatic determination of the number of clusters (K). In this work authors discussed the conceptual advantages of multi-objective clustering and demonstrated that these translate into a performance advantage in practice: the proposed evolutionary approach has been shown to outperform traditional single-objective clustering techniques and an ensemble method across a diverse range of benchmark data sets.

III. Fuzzy Clustering: a brief overview

In clustering (also known as exploratory data analysis), a set of patterns, usually vectors in a multi-dimensional space, are organized into coherent and contrasted groups, such as that patterns in the same group are similar in some sense and patterns in different groups are dissimilar in the same sense. Given a data set of N patterns $X = \{\mathbf{x}_1, ..., \mathbf{x}_i, ..., \mathbf{x}_N\}$, the purpose of any clustering technique is to evolve a partition matrix $U(X)$ of the given data set X so as to find a number, say R, of clusters ($\{U_1, ..., U_R\}$). The partition matrix $U(X)$ of size $R \times n$ may be represented as $U = [u_{ij}]$, $j = 1, .., R$ and $i = 1, ..., n$, where u_{ij} is the membership of pattern $\mathbf{x}_i$ to cluster U_j. In crisp partitioning of the data, the following condition holds: $u_{ij} = 1$ if $\mathbf{x}_i \in U_j$, otherwise $u_{ij} = 0$. In the case of fuzzy clustering, the purpose is to evolve an appropriate partition matrix $U = [u_{ij}]$ where $u_{ij} \in [0, 1]$, such that u_{ij} denotes the degree of membership of the i-th pattern to the j-th cluster.

In pattern recognition fuzzy models and algorithms have been widely studied and applied [25], [26], [27]. In particular one of the major techniques in pattern recognition is fuzzy clustering, that attracts attention because it has been successful in a variety of substantive areas [28], [29], [30], [31] including image recognition, signal processing, business, health, aerospace, and so on.

The Fuzzy C-Means (FCM) is the most famous fuzzy clustering algorithm, which proposes to minimize the following objective function with respect to fuzzy memberships $U = [u_{ij}]$ and cluster centroids $C = [\mathbf{c}_j]$:

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

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 the number of clusters and the number of patterns respectively. 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 an hard partitioning (i.e. u_{ij} becomes a binary variable which is equal to 1 if the i-th pattern belongs to the j-th group, otherwise it is 0).

As distance measure a weighted Euclidean distance was used in this work:

$$d(\mathbf{x}, \mathbf{y}) = \sqrt[2]{\sum_{k=1}^{D} w_k^2 (x_k - y_k)^2} \quad (2)$$

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 data is also to be normalized so as to remove any numerical differences between the features and allow the algorithm to estimate their weights more efficiently. The normalization method we applied in this work was division by the maximum number: X_{norm} = X / X_{max}.

The procedure of evaluating the results of a clustering algorithm is known under the term cluster validity. One of the most effective cluster validity measure is the XB index, that is defined as the ratio of compactness σ/N of the total variation to the minimum separation sep of the clusters, where σ and sep can be written as

$$\sigma(U, C, X) = \sum_{j=1}^{K} \sum_{i=1}^{N} u_{ij}^m d(\mathbf{x}_i - \mathbf{c}_j) \quad (3)$$

and

$$sep(C) = \min_{j \neq k} \{d(\mathbf{c}_j - \mathbf{c}_k)\} \quad (4)$$

$d(\bullet, \bullet)$ is (2). The XB index is then defined as

$$XB(U, C, X) = \frac{\sigma(U, C, X)}{N \cdot sep(C)} \quad (5)$$

Note that, when the partitioning is compact and good, the value of σ should be low, while sep should be high. Therefore, the XB index should have a low value when the data has been appropriately clustered. The presence of m in σ, that is defined as suggested in [7], assures that it has not bias due variability of m. We remark that by definition the XB index has no dimensionality bias too, because it is $sigma$ divided to sep and $\sigma, sep \propto d$ then the XB index is not proportional to d.

IV. Multi-objective optimization: the Nondominated Sort Genetic Algorithm II

The multi-objective optimization problem is a problem of minimization or maximization of multiple evaluation criteria that conflict with each other. It is difficult to say that the solution that is an optimum for one criterion is the optimal solution for multi-objective optimization, because the multiple criteria have trade-off relationships with each other. Therefore, in multi-objective optimization, the concept of a Pareto-optimal solution is used in the search. In a Pareto optimal solution, there are multiple, or sometimes an infinite number of solutions. In multi-objective optimization, as it is mentioned, getting a Pareto-optimal solution is one of the goals and an approach to obtain a wide range of Pareto optimal solutions at equal intervals is required. Evolutionary algorithms are well-suited for multi-objective optimization as their use of a population enables the whole Pareto front to be approximated in a single algorithm run.

NSGA [32] is a popular non-domination based genetic algorithm for multi-objective optimization. It is a very effective algorithm but has been generally criticized for its computational complexity, lack of elitism and for choosing the optimal parameter value for sharing parameter σ_{share}. A modified version, NSGA-II [33] was developed, which has a better sorting algorithm, incorporates elitism and no sharing parameter needs to be chosen a priori. NSGA-II varies from the NSGA (Non-dominated sorting genetic algorithm) in three main things. It is more efficient computationally, since the ranking of solutions is performed by an O(ωM^2) algorithm, instead than O(ωM^3), where ω is the number of objectives and M is the population size; it significantly prevents the loss of good solutions once they have been found (elitism); it does not need any parameter specification. Because of its simplicity, availability of a freely downloadable computer code, and demonstrated superiority over other existing methods, NSGA-II has been extensively used in many studies. Because of its broad-based applicability in academia and practice, NSGA-II has been, since its publication, either used as a baseline algorithm to compare with other methods or has been applied to new problems. NSGA-II is a computationally efficient algorithm implementing the idea of a selection method based on classes of dominance of all the solutions. It uses a fast non-dominated ranking algorithm and a parameter-less sharing mechanism for solutions diversification. In this paper, the normalization of the objectives has also allowed to efficiently compare each objective or constraint contribution during crowded comparison and selection as it is detailed in the following. The population is initialized as usual. Once the population in initialized the population is sorted based on non-domination into each front. The first front being completely non-dominant set in the current population and the second front being dominated by the individuals in the first front only and the front goes so on. Each individual in the each front are assigned rank (fitness) values or based on front in which they belong to. Individuals in first front are given a fitness value of 1 and individuals in second are assigned fitness value as 2 and so on. Before selection is performed, the population is ranked on the basis of an individual's non-domination level and, to allow the diversification, a crowding factor is calculated for each solution. The crowding distance is a measure of how close an individual is to its neighbors. Large average crowding distance will result in better diversity in the population. Then a binary tournament selection operator is used to select the offspring population, whereas crossover and mutation operators remain as usual.

V. Proposed approach

We propose the use of a hybrid system, which simultaneously performs data clustering and feature selection, while it is searching for the suitable number of clusters. It uses a closed-loop control mechanism in which the FCM classification algorithm is controlled by NSGA-II by means of a feedback loop. NSGA-II provides the FCM with the optimal parameters that will ensure the best trade-off between the objectives and assigns weights to each feature. When a feature is irrelevant or redundant its weight will be 0; in this way it will be neglected during the clustering process. 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 the FCM and the feature weights are mapped onto a chromosome whose genes are real coded (see Figure 1). The chromosome genes are: the number of clusters K, the FCM fuzzyfier m, and the D feature weights. A systemic representation is given in Figure 2. It should be pointed out that integrated implementation of the two algorithms causes some problems: the FCM algorithm requires random initialization every time it is executed, on which the result depends. It is obviously not possible to operate in this way with the genetic algorithm, because the fitness value would vary from one generation to another. A way to solve this problem is to insert the initial values of the matrix $U^{(0)}$ among the variables of the GA. Hall *et al.* [34] studied the effects of this strategy, concluding that use of a GA caused an increase in computing time of two orders of magnitude as compared with normal execution. The authors proved that 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 our work we chose to run the FCM algorithm ten times for each number of cluster allowed, $m = 2$ and the feature weights equal to 1, then we select the K (e.g. one for each number of cluster allowed) $U^{(0)}$ matrixes which are associated with the highest value of the objective function. The $U^{(0)}$ matrixes selected are used to initialize the FCM during the generations of the GA.

FCM parameters		***D* Feature weights**		
K	***m***	***w*** $_1$	**...**	***w*** $_D$

Fig. 1. Example of the structure of a chromosome

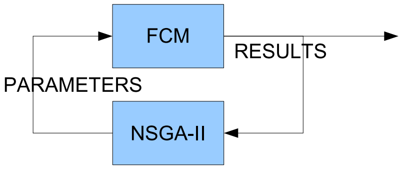

Fig. 2. Systemic representation of the proposed approach.

To summarize, the algorithm is as follows:

1) Initialize the K centroid matrixes by running the FCM 10 times for each number of clusters allowed.
2) Execute NSGA-II to optimize the FCM and to weight the features.
3) Choose the solution most suitable for the problem from the Pareto set solutions.

VI. Numerical Results

The dataset employed is the publicly available Leukemia dataset [29]. This dataset is a particularly difficult testbench because it contains a very low number of patterns while it has a very high number of features. The leukemia problem consists of characterizing two forms of acute leukemia. Acute Lymphoblastic Leukemia (ALL) and Acute Mieloid Leukemia (AML). The original work proposed both a supervised classification task (”class prediction”) and an unsupervised characterization task (”class discovery”). Here we obviously focus on the latter, but we exploit the diagnostic information on the type of leukemia to assess the goodness of the clustering obtained. The dataset contains 38 examples for which the expression level of 7129 genes has been measured with the DNA microarray technique. Of these samples 27 are cases of ALL and 11 are cases of AML. Often, the ALL class is divided into 2 subclasses: the T-lineage and the B-Lineage. For this reason, the suitable number of classes K for this post-genomic dataset is two or three. In this work we use data from [35], where after a series of standard preprocessing steps, the 100 genes with the largest variation across samples were selected, and the remaining expressions are log-transformed. The resulting dataset of size 38×100 is subject to our cluster analysis. The population for the genetic algorithm were set as 200 individuals, using a crossover probability of 0.8 and a mutation probability of 0.1. The stop criterion used for the FCM was the achievement of a maximum variation lower than 0.01 or 15 iterations, whereas for NSGA-II it was 200 generations. The two objectives are the number of features and the XB index, which are both to be minimized. Table I summarizes the parameters search space.

TABLE I
PARAMETERS SEARCH SPACE

Parameter	Parameter Space	Notes
K	[2,10]	integer values
m	[1.01,5.0]	
$w_1 ... w_D$	[0.0,1.0]	We have forced the probability to assign 0.0 to 50% in order to have the same chances either to select or to erase a feature.

Figure 3 shows the values of the Pareto set obtained, which can be divided into two distinct parts: the one on the left that has all the points with $K = 3$ and the one on the right that has all the points with $K = 2$. The results given in Figure 3 confirm that the most suitable number of classes is 2 or 3, because there are no points with $K > 3$ in the Pareto set. The results also suggest that the number of features conflicts not only with performance, but also with the suitable number of clusters. Therefore the 2-class partitioning requires a lower dimensionality but clusters are less compact and separated than in the 3-class partitioning, which vice versa requires a higher dimensionality. However, looking only to XB index, it indicates the higher quality of the three-cluster solution, as reported in the literature. In fact, this three-cluster solution corresponds to an almost perfect separation of the samples of acute leukemia into those arising from myeloid precursors (AML), and two sub-classes arising from lymphoid precursors (T-lineage ALL and B-lineage ALL). m values are only related to the number of clusters, therefore m is in the range [1.01,1.05] for the Pareto points with $K = 2$, while m is in the range [1.28,1.33] for $K = 3$.

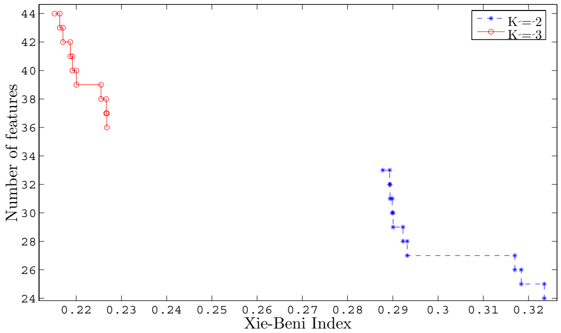

Fig. 3. Pareto set obtained. Points on the left have $K = 3$, while points on the right have $K = 2$.

To evaluate the quality of the partitioning using an external criterion, we built a reference partition using labels of the data, then we derived the crisp (hard) partitions from the optimal fuzzy partitions by assigning each pattern to the group with which it has a higher numerical affinity. These hardened partitions were compared with the reference ones, yielding a maximum classification accuracy of 97% with 34 features and 2 classes and 95% with 44 features and 3 classes. These results are better both in classification accuracy and feature selection than the ones in the original work [36], in which a Self Organizing Map (SOM) is constructed with the *GENECLUSTER* software [36], that reported 89% of accuracy with 2 classes and 95% with 3 classes with 6817 features .

TABLE II
PARTITION ACCURACY COMPARISON (AVERAGES OVER 21 RUNS)

Method	$K = 2$		$K = 3$	
	Features	$AdjRand$	Features	$AdjRand$
NSGA-II & FCM	34	0.96	44	0.93
GENECLUSTER & SOM [36]	6817	0.81	6817	0.93
MOCK & SOTA [35]	100	0.63	100	0.93
MOCK & SOM [35]	100	0.60	100	0.93
MOCK & K-Means [35]	100	0.59	100	0.93
MOCK & Average Link [35]	100	0.52	100	0.93

Note that in [35] no further feature selection was performed, for this reason the number of features is always 100.

In Table II the performance of the proposed approach is compared to the one of the MOCK approach [35], which was ran with various clustering algorithms: self-Organizing Tree Algorithm (SOTA), Self Organizing Maps (SOM), K-Means and the agglomerative hierarchical algorithm based on the average link criteria. In accordance with [35], the Adjusted Rand Index ($AdjRand$) [37], is used as external criterion for the partition quality assessment. The $AdjRand$ index has to be maximized and it can take values in the range $[0, 1]$. Table II shows the average over 21 runs of the number of features needed to achieve the best value for the $AdjRand$ index. In the two cluster scenario the approach proposed gives the best result both in feature selection and in partition quality. In the three cluster scenario all the approaches compared produces a partition with the same quality, but our approach needs less features than others. Note that both *GENECLUSTER* and *MOCK* does not assign weight to the features, this is the reason of their lower performance.

VII. Conclusion

The aim of this paper has been to explore the potential of a fuzzy evolutionary clustering approach based on NSGA-II, integrated with the Fuzzy C-Means algorithm. The new approach presented has proved to be a useful tool for mining knowledge from high-dimensional sets, even in the presence of small sample data. The strength of the proposed algorithm is, therefore, the possibility to discover the best number of groups, which yields the best clustering performance, without a reference classification, while pruning the features in order to reduce the dimensionality of the database. Numerical results confirmed that the right choice of pattern features is a critical issue to achieve high-quality partitioning; in fact growing dimensionality leads to less distinguishability between the groups, but in the meantime a small number of features is not enough to successfully cluster the data. It was also demonstrated that the suitable number of clusters depends on the number of features selected to represent the characteristics of the patterns.

References

[1] A. Jain, M. Murty, and P. Flynn, "Data clustering: A review," *ACM Computing Surveys*, vol. 31, no. 3, 1999.

[2] K. Bayer, J. Goldstein, R. Ramakrishan, and U. Shaft, "When is nearest neightbor meaningful?" in *Proceedings of 7th International Conference on Database Theory (ICDT'99)*, 1999, pp. 217–235.

[3] H. Liu and H. Motoda, *Feature selection for knowledge discovery and data mining*. Kluwer Academic Publishers, 1998.

[4] J. Yang and V. Honavar, "Feature subset selection using a genetic algorithm," *IEEEIntelligent Systems*, vol. 13, no. 2, pp. 44–49, 1998.

[5] H. Liu, E. Dougherty, J. Dy, K. Torkkola, E. Tuv, H. Peng, C. Ding, F. Long, M. Berens, L. Parsons, Z. Zhao, L. Yu, and G. Forman, "Evolving feature selection," *IEEEIntelligent Systems*, vol. 20, no. 6, pp. 64–76, 2005.

[6] X. L. Xie and G. Beni, "A validity measure for fuzzy clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 13, no. 8, pp. 841–847, 1991.

[7] N. R. Pal and J. C. Bezdek, "On cluster validity for the fuzzy c-means model," *Transactions on Fuzzy Systems*, vol. 3, pp. 370–379, Aug. 1995.

[8] R. Babuska, *Fuzzy Modeling for Control*. Dordrecht: Kluwer Academic Press, 1998.

[9] O. Cordn, F. Herrera, F. Hoffmann, and L. Magdalena, *GENETIC FUZZY SYSTEMS Evolutionary Tuning and Learning of Fuzzy Knowledge Bases*, ser. Advances in Fuzzy Systems - Applications and Theory. World Scientific, 2001, vol. 19.

[10] D. van der Merwe and A. Engelbrecht, "Data clustering using particle swarm optimization," in *Proceedings of 2003 Congress on Evolutionary Computation*. IEEE Press, New York, NY, 2003, pp. 215–220.

[11] S. Bandyopadhyay, "Simulated annealing using reversible markov chain monte carlo algorithm for fuzzy clustering," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 4, pp. 479–490, 2005.

[12] S. Nascimento and F. Moura-Pires, "A genetic approach to fuzzy clustering with a validity measure fitness function," in *Proceedings of Second International Symposium on Advances in Intelligent Data Analysis, Reasoning about Data (IDA-97)*, ser. Lecture Notes in Computer Science, X. Liu, P. Cohen, and M. Berthold, Eds., vol. 1280. London, UK: SpringerLink, 1997, pp. 325–335.

[13] T. Van Le, "Evolutionary fuzzy clustering," in *IEEEConf. on Evolutionary Computation (ICEC95)*, vol. 2, Perth, Australia, 1995, pp. 753–758.

[14] B. Yuan, G. J. Klir, and J. F. Swan-Stone, "Evolutionary fuzzy c-means clustering algorithm," in *FUZZ-IEEE '95*, 1995, pp. 2221–2226.

[15] A. Jain and D. Zongker, "Feature selection: Evaluation, application, and small sample performance," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 2, pp. 153–158, 1997.

[16] H. Liu and L. Yu, "Toward integrating feature selection algorithms for classification and clustering," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 4, pp. 491–502, 2005.

[17] R. Kohavi and G. John, "Wrappers for feature subset selection," *Artificial Intelligence*, vol. 97, pp. 273–324, 1997.

[18] F. Hussein, R. K. Ward, and N. N. Kharma, "Genetic algorithms for feature selection and weighting, a review and study," in *ICDAR '01: Proceedings of the Sixth International Conference on Document Analysis and Recognition*, 2001, pp. 1240–1244.

[19] X. Wang, Y. Wangb, and L. Wang, "Improving fuzzy c-means clustering based on feature-weight learning," *Pattern Recognition Letters*, vol. 25, pp. 1123–1132, July 2004.

[20] J. Handl and J. Knowles, "Feature subset selection in unsupervised learning via multiobjective optimization," *International Journal on Computational Intelligence Research*, vol. 3, pp. 217–238, 2006.

[21] A. Ferligoj and V. Bategelj, "Direct multicriterion clustering," *Journal of Classification*, vol. 9, pp. 43–61, 1992.

[22] Y. Kim, W. N. Street, and F. Menczer, "Evolutionary model selection in unsupervised learning," *Intelligent Data Analysis*, vol. 6, no. 6, pp. 531–556, 2002.

[23] M. Morita, R. Sabourin, F. Bortolozzi, and C. Y. Suen, "Unsupervised feature selection using multi-objective genetic algorithms for handwritten word recognition," in *Proceedings of the Seventh International Conference on Document Analysis and Recognition*. IEEE Press, New York, NY, 2003, pp. 666–671.

[24] J. Handl and J. Knowles, "An evolutionary approach to multiobjective clustering," *IEEE Transactions on Evolutionary Computation*, vol. 11, pp. 56–76, Feb. 2007.

[25] A. Baraldi and P. Blonda, "A survey of fuzzy clustering algorithms for pattern recognition. part I-II," *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)*, vol. 29, pp. 778–801, 1999.

[26] J. C. Bezdek, J. Keller, R. Krishnapuram, and N. R. Pal, *Fuzzy Models and algorithms for pattern recognition and image processing*, ser. The handbooks of fuzzy sets series. Springer, 1999.

[27] S. Mitra and S. K. Pal, "Fuzzy sets in pattern recognition and machine intelligence," *Fuzzy Sets and Systems*, vol. 156, pp. 381–386, Dec. 2005.

[28] F. Hppner, F. Klawonn, R. Kruse, and T. Runkler, *Fuzzy Cluster Analysis: Methods for Classification Data Analysis and Image Recognition*. New York: Wiley, 1999.

[29] M. Akay, *Nonlinear Biomedical Signal Processing, Fuzzy Logic, Neural Networks, and New Algorithms*, ser. Series on Biomedical Engineering. IEEE Press, Aug. 2000.

[30] L. C. Jain and M. Sato-ILIC, *Innovations in Fuzzy Clustering: Theory and Applications Theory And Applications*, ser. Studies In Fuzziness And Soft Computing. Springer, Oct. 2006.

[31] J. V. de Oliveira and W. Pedrycz, *Advances in Fuzzy Clustering and its Applications*. New York: Wiley, Mar. 2007.

[32] N. Srinivas and K. Deb, "Multiobjective optimization using nondominated sorting in genetic algorithms," *Evolutionary Computation*, vol. 2, pp. 221–248, 1994.

[33] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multi-objective genetic algorithm: NSGA-II," *IEEE Transaction on Evolutionary Computation*, vol. 6, pp. 181–197, 2002.

[34] L. O. Hall, I. B. Ozyurt, and J. C. Bezdek, "Clustering with a genetically optimized approach," *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 2, pp. 103–112, 1999.

[35] J. Handl, J. Knowles, and D. B. Kell, "Computational cluster validation in post-genomic data analysis," *Bioinformatics*, vol. 21, no. 15, pp. 3201–3212, 2005.

[36] T. Golub, D. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. Mesirov, H. Coller, M. Loh, J. Downing, M. Caligiuri, C. Bloomfield, and L. E.S, "Molecular classification of cancer: class discovery and class prediction by gene expression monitoring," *Science*, vol. 286, pp. 531–537, 1999.

[37] L. Hubert and P. Arabie, "Comparing partitions," *Journal of Classification*, pp. 193–218, 1985.