

A fuzzy clustering method using Genetic Algorithm and Fuzzy Subtractive Clustering

Thanh Le¹, Tom Altman¹, Katheleen J. Gardiner²

¹Department of CSE, University of Colorado Denver, Denver, CO, USA

²Department of Pediatrics, University of Colorado Denver, Aurora, CO, USA

Abstract– Clustering is a challenging problem in data mining, requiring both accurate determination of the number of clusters and correct clustering of the data. Fuzzy C-means (FCM) is a popular algorithm using the partitioning approach to solve this problem. A drawback to FCM is that it requires the number of clusters to be set a priori. In this study, we combine FCM with Genetic Algorithm (GA), Subtractive Clustering (SC) and Bayesian cluster validation for a novel clustering method, fzGASCE that both determines the correct number of clusters and efficiently constructs these clusters from a given dataset. We show that fzGASCE outperforms existing methods using similar approaches on both artificial and real datasets.

Availability: The test datasets and the method software are available online at <http://ouray.ucdenver.edu/~tnle/fzgasce>.

Keywords: fuzzy c-means, genetic algorithm, subtractive clustering, Bayesian cluster validity.

1 Introduction

Clustering in data mining refers to the grouping of data points within a dataset based on their similar properties. Data points within a cluster are highly similar to each other and can be discriminated from data points within other clusters. Successful clustering, therefore, maximizes both the compactness of data points within a cluster and the discrimination between clusters. A clustering problem generally contains two parts: (i) determination of the

This work was supported in part by the Vietnamese Ministry of Education and Training (TL).

Thanh Le is a doctoral student in the Department of Computer Science and Engineering, University of Colorado Denver, Denver, CO 80217, USA (email: lnmail@yahoo.com).

Tom Altman is a professor in the Department of Computer Science and Engineering, University of Colorado Denver, Denver, CO 80217, USA (email: tom.altman@ucdenver.edu).

Katheleen J. Gardiner is a professor in the Department of Pediatrics and the Linda Crnic Institute for Down Syndrome; and a member of the Intellectual and Developmental Disabilities Research Center; and the Computational Biosciences, Human Medical Genetics and Neuroscience Programs, University of Colorado Denver, Aurora, CO 80045, USA (phone: 303-724-0572; email: katheleen.gardiner@ucdenver.edu).

number of clusters and (ii) assignment of data points to clusters. In the simplest application, the data are to be partitioned into only two clusters such that the summation of some non-geometric distance measures between the data points of the clusters is maximized. This clustering problem can be shown to be an NP-hard decision problem, for which there are currently no efficient algorithms that would run in polynomial time with respect to the dataset size [16]. Such problems are therefore “computationally intractable”, thus allowing application of heuristic or approximation algorithms that provide results that may be adequate although not always optimal.

Clustering approaches include partitioning and hierarchical methods. Partitioning approaches include two methods for managing cluster boundaries: the well-known K-means algorithm that uses crisp boundaries and the Fuzzy C-means (FCM) [1] that uses fuzzy cluster boundaries, where fuzzy sets are applied to associate every data point with at least one cluster. The K-means algorithm requires that every data point belongs to only one cluster. It therefore may not be appropriate for some real applications, such as gene expression analysis or pattern recognition. On the other hand, the FCM algorithm, using the concept of fuzzy set theory to allow every data point to belong to more than one cluster, is more appropriate to real-world problems and it has become the most popular partitioning method. However, the FCM algorithm, like most other partitioning approaches, cannot by itself determine the number of clusters, and results depend strongly on initial parameters. For some initial values, FCM will converge rapidly to a global optimum, but, for others, it may become stuck in a local optimum.

One approach to address the limitations of FCM is to integrate FCM with the Genetic Algorithm (GA), where the GA is used to manage a set of solution candidates, the FCM algorithm is applied and a cluster validity index is used as the GA fitness function to search for the best solution. Ghosh et al. [7] and Liu et al. [10] proposed to use the partition coefficient (PC) [1] and Xie and Beni (XB) [14] validity indices. In addition, Liu et al. [10] proposed a modified version of the PC index (MPC) in order to reduce the monotonic tendency of the index. The Fukuyama-Sugeno cluster index (FS) [5], which measures the

compactness and separation of the cluster partition, was used by Ghosh et al. [7]. Lianjiang et al. [8], in a novel self-Adaptive Genetic Fuzzy C-Means algorithm (AGFCM), proposed a validity index combining the PC index with total variation of fuzzy partition. Halder, Pramanik and Kar [6] proposed a GA fitness function (HPK) based on the compactness measure and combined it with an intra-inter validity index for a novel algorithm that automatically determines the number of clusters in the dataset. Lin et al. [9] proposed a combination of GA and FCM for a novel method with adaptive cluster validity index (ACVI) based on the intra and inter measures of the fuzzy partition, where GA is used to solve the trade-off problem between these two factors for a better evaluation of the fuzzy partition.

A common limitation of existing methods using GA with FCM is that the GA fitness functions are based on the cluster validity indices, which usually have a problem with scaling between the compactness and separation factors. In addition, they use the maximum membership degree for defuzzification that may be improper, because the membership is computed based on the distance between the data object and cluster center. Use of membership degree can assign marginal objects of a large cluster to the immediately adjacent small cluster. This is illustrated in Figure 1, where if a data object is in the gray rectangle, it may be incorrectly assigned to cluster 3 instead of cluster 2.

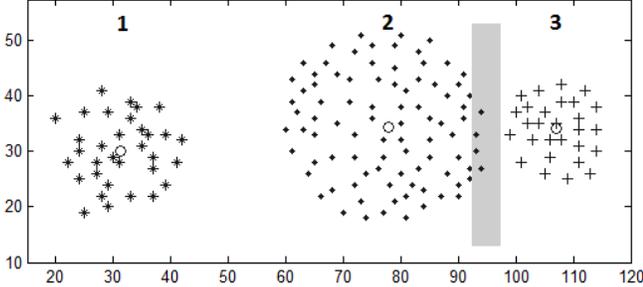


Figure 1: ASET4 - an artificial dataset with three clusters of different sizes

In this study, we combine FCM with GA and fuzzy SC algorithms for a novel clustering algorithm, fzGASCE, that automatically determines the number of clusters in the dataset. The FCM algorithm rapidly determines the exact clustering prototypes of each solution candidate so that the GA algorithm, managing a set of such candidates, can select the optimal one. The fuzzy SC method helps the GA algorithm escape any local optima.

2 Methods

2.1 Fuzzy C-Means algorithm (FCM)

Let $X = \{x_1, x_2, \dots, x_n\} \in \mathbb{R}^p$ be a set of data objects x_i , $i=1..n$. For a given c , $2 \leq c < n$, the Fuzzy C-Means algorithm

(FCM) divides X into c clusters by minimizing the objective function:

$$J_m(U, V) = \sum_{i=1}^n \sum_{k=1}^c u_{ki}^m d^2(x_i, v_k) \rightarrow \min \quad (1)$$

where $u_{ki} \in [0, 1] \forall k, i$,

$$\sum_{k=1}^c u_{ki} = 1 \forall i, \quad (2)$$

and m , $1 \leq m$, is the fuzzifier factor; $V, V = \{v_1, v_2, \dots, v_c\}$ is a set of c cluster centers; $U = \{u_{ki}\}_{i=1..n, k=1..c}$ is a partition matrix; and $d^2(\cdot)$ denotes the Euclidean norm.

Minimizing J_m with respect to (2), we obtain an estimated model of U and V as:

$$u_{ki} = \left(\frac{1}{d^2(x_i, v_k)} \right)^{\frac{1}{m-1}} / \sum_{l=1}^c \left(\frac{1}{d^2(x_i, v_l)} \right)^{\frac{1}{m-1}}, \quad (3)$$

$$v_k = \sum_{i=1}^n u_{ki}^m x_i / \sum_{i=1}^n u_{ki}^m. \quad (4)$$

FCM can converge rapidly and provide soft partitions applicable to many real-world applications. However, FCM depends strongly on initial parameters and can become stuck in a local optimum.

2.2 Genetic Algorithms (GA)

Genetic algorithms (GAs) are a class of optimization algorithms that perform adaptive searches to find solutions to large scale optimization problems with multiple local optima. When applied to cluster analysis, conventional GAs use chromosomes to describe solution candidates. Each chromosome represents a part or the whole of a solution. In the case of the later, a chromosome is a set of loci each standing for a data point selected as the center of a cluster in the solution. GAs manage a set of chromosomes to search for the best solution through an evolution process. At each generation, new offspring are created from the parents, chosen based on either tournament or roulette wheel selection methods, using the crossover and mutation operators. The worst members in the current generation will be replaced by the newly created ones to construct the next generation.

The crossover and mutation operators help GAs to escape local optima. These operators depend strongly on how the probabilities of crossover and mutation are chosen. Recent improvements in GAs have focused on adaptively adjusting operator probabilities so that the genetic processes rapidly escape local optima. However, setting up adaptive GAs is difficult and most approaches are based on heuristics.

3 The fzGASCE algorithm

We propose a novel fuzzy clustering algorithm, fzGASCE, which combines the GA and fuzzy SC algorithms with a Bayesian based cluster evaluation method to overcome the drawbacks of existing methods that use the GA and FCM algorithms.

Chromosome

We used chromosome to represent the whole clustering solution. Each chromosome contains a set of loci each standing for the index of the data point selected as cluster center. We set the length of chromosomes to \sqrt{n} , which is assumed the maximum number of clusters in the dataset.

Crossover operator

The crossover operator is used to produce two new offspring from a given pair of parents. Both the roulette wheel and tournament selection methods are used interchangeably to select parents maintaining potentially useful solutions in the current generation. A two-point crossover operator with probability P_c , $P_c = 0.5$, is used.

Mutation operator

The mutation operator is used to make changes in portions of the chromosomes of newly created members. Because each chromosome encodes data points representing the cluster centers of a clustering solution, changing of the data points in the chromosome may improve the clustering quality. We therefore propose three different tasks for the mutation operator: (i) add a data point as a new cluster center, because this may help to locate a new cluster in a higher density region, (ii) remove a data point to prevent the inclusion of a sparse cluster, and (iii) replace one data point with another so that the new cluster is located in a higher density region. These tasks are commonly used in existing methods. However, they are employed in a random or heuristic way, and cannot guarantee that the GA algorithm will escape local optima. To address this issue, we propose an alternative approach using the fuzzy SC method of Le et al. [13]. The fuzzy partitions of the parents are used to estimate the density at every data point.

$$\text{dens}(x_i) = \sum_{k=1}^c \text{Acc}(v_k) \times u'_{ki}, \quad (5)$$

where, $\text{Acc}(v_k)$, the accumulated density at v_k , and $\{u'_{ki}\}$, a strong uniform fuzzy partition of $\{u_{ki}\}$ [2], are defined, respectively, for $k=1..c$, $i=1..n$ as:

$$\text{Acc}(v_k) = \sum_{i=1}^n u_{ki}, \quad (6)$$

$$u'_{ki} = \left[e^{d(x_i, v_k)/\sigma_k} \right]^{-1} / \sum_{l=1}^c \left[e^{d(x_i, v_l)/\sigma_l} \right]^{-1}. \quad (7)$$

At each time, t , the data point, x_t^* with the highest density, M_t^* , is selected, and the densities at the remaining data points are updated as:

$$\text{dens}^{t+1}(x_i) = \text{dens}^t(x_i) - M_t^* \sum_{k=1}^c u'_{ki} \times P(v_k | x_t^*). \quad (8)$$

The points where the densities in this way change less than a predefined ratio, R_M , are considered to be significantly dense and are used in the mutation operator of fzGASCE. This is instead of using randomly selected data points, as in existing methods. We chose a value of 0.95 for R_M . The selection of the value of R_M does not affect the outcome however a low value of R_M may slow the convergence process of fzGASCE.

Fitness function

Instead of using a cluster validity index for the fitness function, we use the method of Le et al. [12] for cluster evaluation. For each chromosome, FCM is applied to generate the fuzzy partition which is then used to generate a probabilistic model of the data distributions [3, 12]. For each cluster v_k , $k=1..c$, the probability distribution $\{p_{ki}\}_{i=1..n}$ is derived from the possibility distribution $\{u_{ki}\}_{i=1..n}$. Then, the following statistics at v_k are computed:

$$\sigma_k = \sum_{i=1}^n p_{ki} \|x_i - v_k\|^2, \quad (9)$$

$$P(v_k) = \frac{\sum_{i=1}^n P(x_i | v_k)}{\sum_{l=1}^c \sum_{i=1}^n P(x_i | v_l)}, \quad (10)$$

$$P(x_i | v_k) = \left((2\pi)^{1/n} \times \sigma_k \times e^{-\frac{\|x_i - v_k\|^2}{2\sigma_k^2}} \right)^{-1}, \quad (11)$$

where σ_k and $P(v_k)$ are the variance and the prior probability of v_k respectively; $P(x_i|v_k)$ indicates the conditional probability of x_i given v_k , for $i=1..n$, $k=1..c$.

Given a fuzzy partition θ , $\theta = \{U, V\}$, the fitness function is defined:

$$\text{fitness}(\theta) = \log[L(\theta|X)] - \log(c), \quad (12)$$

where $L(\theta|X)$, the likelihood of the clustering model and the data, is measured as:

$$L(\theta | X) = L(U, V | X) \\ = \prod_{i=1}^n P(x_i | U, V) = \prod_{i=1}^n \sum_{k=1}^c P(v_k) \times P(x_i | v_k). \quad (13)$$

Defuzzification of fuzzy partition

Use of the maximum fuzzy membership degree to determine classification of data points result in assignment of marginal objects of a large cluster to the immediately adjacent small cluster. The gray rectangle in Figure 1 shows an example where data points may be incorrectly assigned to cluster 3 instead of to cluster 2. Therefore, we use the probabilistic model. A data point $x_i, i=1..n$, will be assigned to cluster $v_k, k=1..c$, where:

$$P(v_k | x_i) = \max_{i=1..c} \{P(v_i | x_i)\}. \quad (14)$$

Because $P(v_k | x_i) = P(x_i, v_k) / P(x_i) = P(x_i | v_k) * P(v_k) / P(x_i)$, we solve this as:

$$P(v_k | x_i) = \max_{i=1..c} \{P(x_i | v_i) \times P(v_i)\} \quad (15)$$

fzGASCE algorithm

- Input: data to cluster $X = \{x_i\}, i=1..n$.
- Output: an optimal fuzzy clustering solution,
 - c : optimal number of clusters.
 - $V = \{v_i\}, i=1..c$: the cluster centers.
 - $U = \{u_{ki}\}, i=1..n, k=1..c$: the partition matrix.

Steps

1. Randomly generate a set of chromosomes describing clustering solution candidates.
2. Compute the fitness value for every chromosome.
3. If the stop criteria are met, then go to step 7.
4. Apply the crossover operator with the probability $P_c = 0.5$, and the roulette wheel and tournament parent selection methods.
5. Apply the mutation operator with the probability $P_m = 0.01$. The significantly dense data points are used in the replacement, and fresh offspring are created using these points with a probability of $P_c \times P_m$.
6. Go to step 2.
7. Select the 'best' chromosome from the population as the clustering solution.
8. Apply (15) for the defuzzification of the fuzzy partition of the solution.

4 Experimental results

Datasets

To evaluate the performance of fzGASCE, we used four artificial datasets generated using an infinite mixture

model method [15]. ASET1, ASET2 and ASET3 each have well-separated clusters of similar sizes. The number of clusters and data dimensions of these dataset are (5,2), (5,3) and (11,5), respectively. ASET4 is more complex, containing three clusters that differ in size and density (Figure 1). For real datasets, we used the Iris and Wine datasets from the University of California Irvine (UCI) Machine Learning Repository [4]. The classification structures in these datasets are known.

Performance measures

We used three measures, COR, EVAR and EMIS, to evaluate algorithm performance. COR is the correctness ratio, defined as,

$$COR = \frac{1}{N} \sum_{i=1}^N I(c - \hat{c}), \quad (16)$$

where N is the number of trials, c and \hat{c} are the number of clusters and the predicted number of clusters respectively, and $I(\cdot)$ is defined as:

$$I(x) = \begin{cases} 1, & x = 0 \\ 0, & x \neq 0 \end{cases}. \quad (17)$$

EVAR is a measure of the accuracy of the predicted number of clusters defined as in (18).

$$EVAR = \frac{1}{N} \sqrt{(c - \hat{c})^2}. \quad (18)$$

EMIS is a measure of the overall performance, determined by the number of data objects that were misclassified. EMIS is calculated only when an algorithm correctly identifies the number of clusters. Then, the assigned cluster label of each object is compared with its actual cluster label. If any of them do not match, a misclassification has occurred

We compared performance of fzGASCE with eight genetic algorithm methods that also use FCM, specifically, PBMF, MPC, HPK, AGFCM, XB, FS, PC and ACVI [6-10]. An earlier version of fzGASCE, fzGAE which does not include the fuzzy SC and defuzzification methods, was also used. For each dataset, the number of clusters, c , was set to the known number of clusters. All algorithms were run using a population size of 24 and a maximum number of 100 generations. The fuzzy partition of each chromosome was generated using the FCM algorithm with 10 iterations, and the fuzzifier factor, m , was set to 1.25. We repeated the experiment 100 times and averaged the performance of each algorithm using values of COR, EVAR and EMIS.

ASET1 dataset

ASET1 contains five clusters in a 2-dimensional data space. The clusters are well-separated and of the same size. Performance of all algorithms is shown in Table 1. All algorithms had very low EMIS measures, indicating that they grouped the data points into the correct clusters. However, fzGASCE outperformed all algorithms by all three measures and fzGAE performed better than the other methods with the exception of fzGASCE. This comparison illustrates the advantage of using the method of Le et al. [12] in the fitness function. The use of fuzzy SC [13] in fzGASCE improved performance particularly in escaping local optima.

Table 1
Results with ASET1

Algorithm	COR	EVAR	EMIS
fzGASCE	1.000	0.000	0.000
fzGAE	0.640	0.500	0.000
PBMF	0.510	0.590	0.000
MPC	0.290	0.970	0.000
HPK	0.100	5.010	0.021
AGFCM	0.600	2.800	0.000
XB	0.490	1.450	0.000
FS	0.120	1.100	0.070
PC	0.230	1.040	0.000
ACVI	0.200	2.490	0.011

ASET2 and ASET3 datasets

Table 2
Results with ASET2

Algorithm	COR	EVAR	EMIS
fzGASCE	1.000	0.000	0.000
fzGAE	0.710	0.380	0.000
PBMF	0.600	0.450	0.000
MPC	0.610	0.860	0.000
HPK	0.120	5.240	0.000
AGFCM	0.650	1.490	0.000
XB	0.640	0.430	0.000
FS	0.520	0.840	0.011
PC	0.620	0.890	0.000
ACVI	0.100	2.100	0.000

Table 3

Results with ASET3

Algorithm	COR	EVAR	EMIS
fzGASCE	1.000	0.000	0.000
fzGAE	0.450	0.750	0.000
PBMF	0.340	1.000	0.000
MPC	0.420	0.820	0.002
HPK	0.010	1.910	0.037
AGFCM	0.340	2.380	0.000
XB	0.410	0.900	0.000
FS	0.400	0.880	0.000
PC	0.450	0.700	0.000
ACVI	0.170	4.650	0.000

The ASET2 contains five well-separated clusters in a 3-dimensional data space while the ASET3 contains 11 clusters in a 5-dimensional data space. Performance of all algorithms on ASET2 and ASET3 are shown in Tables 2 and 3 respectively. On both datasets, fzGASCE outperformed all other algorithms, while fzGAE was the second at best.

ASET4 dataset

This dataset is non-uniform with three clusters in a 2-dimensional data space (Figure 1). Table 4 shows the algorithm performance. HPK and AGFCM both failed to determine the number of clusters. fzGASCE not only outperformed the other algorithms, with COR=1.0, but also was the only algorithm that grouped all data points into the correct clusters. Although fzGAE performed better than the remaining algorithms, it failed to correctly group data points into clusters, similar to other methods that used the maximum membership degree for defuzzification.

Table 4

Algorithm performance on the ASET4 dataset

Algorithm	COR	EVAR	EMIS
fzGASCE	1.000	0.000	0.000
fzGAE	0.900	0.100	0.107
PBMF	0.700	0.300	0.107
MPC	0.050	0.960	0.107
HPK	0.000	5.770	-
AGFCM	0.000	8.470	-
XB	0.040	0.960	0.107
FS	0.020	3.480	0.107
PC	0.050	0.960	0.107
ACVI	0.080	0.920	0.107

IRIS dataset

The IRIS dataset contains three clusters corresponding to the three classes of Iris flowers [4]. The performance of the algorithms on this dataset is shown in Table 5. HPK and AGFCM again completely failed at detecting the number of clusters. fzGASCE outperformed other algorithms in detecting the number of clusters as well as in grouping data points into their own clusters.

Table 5
Algorithm performance on the IRIS dataset

Algorithm	COR	EVAR	EMIS
fzGASCE	1.000	0.000	0.033
fzGAE	0.880	0.120	0.040
PBMF	0.860	0.140	0.040
MPC	0.040	0.970	0.160
HPK	0.000	5.720	-
AGFCM	0.000	8.120	-
XB	0.050	1.010	0.040
FS	0.390	0.780	0.154
PC	0.080	0.920	0.115
ACVI	0.150	0.850	0.040

Wine dataset

Table 6
Algorithm performance on the WINE dataset

Algorithm	COR	EVAR	EMIS
fzGASCE	1.000	0.000	0.213
fzGAE	0.860	0.140	0.303
PBMF	0.000	2.050	-
MPC	0.000	2.810	-
HPK	0.000	6.760	-
AGFCM	0.000	9.210	-
XB	0.270	1.010	0.303
FS	0.000	5.720	-
PC	0.110	0.920	0.303
ACVI	0.090	0.910	0.303

The Wine dataset contains information on 13 attributes of three classes of wines [4]. Results on this dataset are shown in Table 6. Only fzGASCE and fzGAE identified the correct number of clusters, with COR values of 1 and 0.86, respectively. Among the other algorithms, only XB and ACVI detected the correct number of clusters but only with low COR values. Overall, fzGASCE outperformed all of the tested algorithms.

5 Conclusions

We have presented fzGASCE, a novel fuzzy clustering algorithm that combines the Genetic Algorithm with the fuzzy subtractive clustering and Bayesian based cluster evaluation methods. fzGASCE solves the problem of data clustering in the absence of information on the real number of clusters. fzGASCE outperformed other methods on both artificial and real datasets, and performed particularly well on datasets with clusters that differed in size, not only in predicting the correct number of clusters but also in grouping the data points into the correct clusters. fzGASCE is therefore appropriate for real-world problems, where the data densities are not uniformly distributed. In future work, we will develop fzGASCE into a more powerful tool for cluster analysis of microarray gene expression data.

6 References

- [1] J.C. Bezdek, "Pattern Recognition with Fuzzy Objective Function Algorithms", Plenum Press, New York, 1981.
- [2] K. Loquin and O. Strauss, "Histogram density estimators based upon a fuzzy partition", Statistics and Probability Letters, Vol. 78, pp. 1863–1868, 2008.
- [3] M.C. Florea, A.L. Joussemme, D. Grenier, and E. Bosse, "Approximation techniques for the transformation of fuzzy sets into random sets", Fuzzy Sets and Systems, Vol. 159, pp. 270–288, 2008.
- [4] A. Frank and A. Asuncion, (2010) Machine Learning Repository. [Online]. <http://archive.ics.uci.edu/ml>.
- [5] Y. Fukuyama and M. Sugeno, "A new method of choosing the number of clusters for the fuzzy c-means method", in: Proc. Fifth Fuzzy Systems Symp., 1989, pp. 247–250.
- [6] A. Halder, S. Pramanik, and A. Kar, "Dynamic Image Segmentation using Fuzzy C-Means based Genetic Algorithm," International Journal of Computer Applications, Vol. 28, pp. 15-20, 2011.
- [7] A. Ghosh, N. S. Mishra, and S. Ghosh, "Fuzzy clustering algorithms for unsupervised change detection in remote sensing images," Information Sciences, Vol. 181, pp. 699-715, 2011.
- [8] Z. Lianjiang, Q. Shouning, and D. Tao, "Adaptive fuzzy clustering based on genetic algorithm," In Proc. of 2nd conference on advanced computer control, Shenyang China, 2010, pp. 79-82.
- [9] T.C. Lin, H.C. Huang, B.Y. Liao, and J.S. Pan, "An Optimized Approach on Applying Genetic Algorithm to

Adaptive Cluster Validity Index,” International Journal of Computer Sciences and Engineering Systems, Vol. 1, pp. 253-257, 2007.

[10] Y. Liu and Y. Zhang, “Optimizing Parameters of Fuzzy c-Means Clustering Algorithm,” In Proc. of the 4th conference on Fuzzy Systems and Knowledge Discovery (FSKD '07), Vol. 1, 2007, pp. 633-638.

[11] M.K. Pakhira, S. Bandyopadhyay, and U. Maulik, “Validity index for crisp and fuzzy clusters”, Pattern Recognition, Vol. 37, pp. 481–501, 2004.

[12] Thanh Le and Katherine Gardiner, “A validation method for fuzzy clustering of gene expression data,” Proc. Intl' Conf. on Bioinformatics and Computational, Las Vegas USA, 2011, Vol. 1, pp. 23-29.

[13] Thanh Le and Tom Altman, “A new initialization method for the Fuzzy C-Means Algorithm using Fuzzy Subtractive Clustering,” Proc. Intl' Conf. on Information and Knowledge Engineering, Las Vegas USA, 2011, Vol. 1, pp. 144-150.

[14] X.L. Xie and G. Beni, “A validity measure for fuzzy clustering”, IEEE Trans. Pattern Anal. Mach. Intell, Vol. 13, pp. 841–847, 1991.

[15] L. Xu and M.I. Jordan, “On convergence properties of the EM algorithm for Gaussian mixtures”, Neural Computation, Vol. 8, pp. 129–151, 1996.

[16] M. Garey and D. Johnson, “Computers and Intractability- A Guide to the Theory of NP-completeness,” Freeman, 1979.