Genetic Based Interval Type-2 Fuzzy C-Means Clustering

Dzung Dinh Nguyen, Long Thanh Ngo, and Long The Pham

Department of Information Systems, Faculty of Information Technology,
Le Quy Don Technical University, No 100, Hoang Quoc Viet, Hanoi, Vietnam
{dinhdung1082,ngotlong}@gmail.com

Abstract. This paper deals with a genetic-based interval type 2 fuzzy c-means clustering (GIT2FCM), which automatically find the optimal number of clusters. A heuristic method based on a genetic algorithm (GA) is adopted to automatically determine the number of cluster based on the validity index. The proposed algorithm contains two main steps: initialize randomly the population of the GA and use the GA to adjust the cluster centroids based on the validity index which is computed by interval type 2 fuzzy c-means clustering (IT2FCM). The experiments are done based on datasets with the statistics show that the algorithm generates good quality of clusters.

Keywords: fuzzy clustering, interval type-2 fuzzy c-means, genetic algorithm.

1 Introduction

Clustering is used to detect any structures or patterns in the data set, in which objects within the cluster level data show certain similarities. Clustering algorithms have different shapes from simple clustering as k-means and various improvements [2,3,4,5], development of family of fuzzy c-mean clustering (FCM) [11]. In addition, genetic algorithms (GAs) have been proposed as alternatives to carry out optimization search due to its prosperities of multi-objective, coded variables and global optimization. The GA [20] is an artificial genetic system based on the principle of natural selection where stronger individuals are likely the winners in a competing environment. GA as a tool for search and optimization has reached a mature stage with the development of low cost and speedy computers. Thus, many studies have proposed an algorithm which integrated GA and FCM called genetic fuzzy c-mean algorithm (GFCM). GFCM has been successfully applied to several data analysis problems such as image processing [24], bio informatics [23]...

However, the general clustering algorithms and GFCM often have difficulty in determining the number of clusters and choosing the initial centroids of the clusters. They have limitation in the handling of the uncertainty and the combination between FCM and GA. Recently, type-2 fuzzy sets are extensions of original fuzzy sets, have the advantage of handling uncertainty, which have been developed and applied to many different problems [6,7,8,9] including data clustering problems. In addition, interval type-2 fuzzy c-means clustering (IT2FCM) [1] has developed a step in the clustering

P.C. Vinh et al. (Eds.): ICCASA 2012, LNICST 109, pp. 239–248, 2013.

method in which FOU (footprint of uncertainty) is created for the fuzzier m using two parameters for handling of uncertainty, making clustering more efficiently

Therefore, This paper proposes a genetic interval type 2 fuzzy c-mean clustering algorithm (GIT2FCM) based on a validity index to automatically find the optimal initial centroids of the clusters and determine the number of clusters. The proposed algorithm uses the cluster validity measure proposed by Ramze Rezaee [10] to evaluate the clustering results. By minimizing this validity measure by GA through adjusting the initial centroids of the clusters and the number of clusters, the proposed algorithm will find the optimal number of clusters. The algorithm consists of two steps. The first step, we randomly initiate the population of the GA. The second step, a GA adjusts the cluster centroids based on the validity index as a fitness function which is computed by IT2FCM. The proposed algorithm also automatically determine the optimal number of clusters. Experiments are implemented based various datasets of classification to show the advantage of proposed approach.

Remain of the paper is organized as follows: Section II briefly introduces about type-2 fuzzy sets, IT2FCM clustering and GA. Section III describes the proposed algorithm; Section IV offers some experimental results and section V concludes the paper.

2 Preliminaries

2.1 Type-2 Fuzzy Sets

A type-2 fuzzy set [6] in X is denoted $\tilde{A}$, and its membership grade of $x \in X$ is $\mu_{\tilde{A}}(x, u), u \in J_x \subseteq [0, 1]$, which is a type-1 fuzzy set in [0, 1]. The elements of domain of $\mu_{\tilde{A}}(x, u)$ are called primary memberships of x in $\tilde{A}$ and memberships of primary memberships in $\mu_{\tilde{A}}(x, u)$ are called secondary memberships of x in $\tilde{A}$.

Definition 1. *A type-2 fuzzy set, denoted $\tilde{A}$, is characterized by a type-2 membership function $\mu_{\tilde{A}}(x, u)$ where $x \in X$ and $u \in J_x \subseteq [0, 1]$, i.e.,*

$$\tilde{A} = \{((x, u), \mu_{\tilde{A}}(x, u)) | \forall x \in X, \forall u \in J_x \subseteq [0, 1]\} \tag{1}$$

or

$$\tilde{A} = \int_{x \in X} \int_{u \in J_x} \mu_{\tilde{A}}(x, u))/(x, u), J_x \subseteq [0, 1] \tag{2}$$

in which $0 \leq \mu_{\tilde{A}}(x, u) \leq 1$.

At each value of x, say $x = x'$, the 2-D plane whose axes are u and $\mu_{\tilde{A}}(x', u)$ is called a $vertical\ slice$ of $\mu_{\tilde{A}}(x, u)$. A *secondary membership function* is a vertical slice of $\mu_{\tilde{A}}(x, u)$. It is $\mu_{\tilde{A}}(x = x', u)$ for $x \in X$ and $\forall u \in J_{x'} \subseteq [0, 1]$, i.e.

$$\mu_{\tilde{A}}(x = x', u) \equiv \mu_{\tilde{A}}(x') = \int_{u \in J_{x'}} f_{x'}(u)/u, J_{x'} \subseteq [0, 1] \tag{3}$$

in which $0 \leq f_{x'}(u) \leq 1$.

Type-2 fuzzy sets are called an interval type-2 fuzzy sets [8] if the secondary membership function $f_{x'}(u) = 1 \ \forall u \in J_x$ i.e. a type-2 fuzzy set are defined as follows:

Definition 2. *An interval type-2 fuzzy set $\tilde{A}$ is characterized by an interval type-2 membership function $\mu_{\tilde{A}}(x,u) = 1$ where $x \in X$ and $u \in J_x \subseteq [0,1]$, i.e.,*

$$\tilde{A} = \{((x,u),1) | \forall x \in X, \forall u \in J_x \subseteq [0,1]\} \tag{4}$$

Uncertainty of $\tilde{A}$, denoted FOU, is union of primary functions i.e. $FOU(\tilde{A}) = \bigcup_{x\in X} J_x$. Upper/lower bounds of membership function (UMF/LMF), denoted $\overline{\mu}_{\tilde{A}}(x)$ and $\underline{\mu}_{\tilde{A}}(x)$, of $\tilde{A}$.

2.2 Interval Type-2 Fuzzy Clustering Algorithm

IT2FCM is extension of FCM clustering by using two fuzziness parameters m_1, m_2 to make FOU, corresponding to upper and lower values of fuzzy clustering (detail in [1]). The use of fuzzifiers gives different objective functions to be minimized as follows:

$$\begin{cases} J_{m_1}(U,v) = \sum_{k=1}^{N} \sum_{i=1}^{C} (u_{ik})^{m_1} d_{ik}^2 \\ J_{m_2}(U,v) = \sum_{k=1}^{N} \sum_{i=1}^{C} (u_{ik})^{m_2} d_{ik}^2 \end{cases} \tag{5}$$

in which $d_{ik} = \| x_k - v_i \|$ is Euclidean distance between the pattern x_k and the centroid v_i, C is number of clusters and N is number of patterns. Upper/lower degrees of membership, $\overline{u}_{ik}$ and $\underline{u}_{ik}$ are determined as follows:

$$\overline{u}_{ik} = \begin{cases} \dfrac{1}{\sum_{j=1}^{C} (d_{ik}/d_{jk})^{2/(m_1-1)}} \; \textit{if} \; \dfrac{1}{\sum_{j=1}^{C} (d_{ik}/d_{jk})} < \dfrac{1}{C} \\ \dfrac{1}{\sum_{j=1}^{C} (d_{ik}/d_{jk})^{2/(m_2-1)}} \; \textit{if} \; \dfrac{1}{\sum_{j=1}^{C} (d_{ik}/d_{jk})} \geq \dfrac{1}{C} \end{cases} \tag{6}$$

$$\underline{u}_{ik} = \begin{cases} \dfrac{1}{\sum_{j=1}^{C} (d_{ik}/d_{jk})^{2/(m_1-1)}} \; \textit{if} \; \dfrac{1}{\sum_{j=1}^{C} (d_{ik}/d_{jk})} \geq \dfrac{1}{C} \\ \dfrac{1}{\sum_{j=1}^{C} (d_{ik}/d_{jk})^{2/(m_2-1)}} \; \textit{if} \; \dfrac{1}{\sum_{j=1}^{C} (d_{ik}/d_{jk})} < \dfrac{1}{C} \end{cases} \tag{7}$$

in which $i = \overline{1,C}$, $k = \overline{1,N}$.

Because each pattern has membership interval as the upper $\overline{u}$ and the lower $\underline{u}$, each centroid of cluster is represented by the interval between v^L and v^R. Cluster centroids is computed in the same way of FCM as follows:

$$v_i = \frac{\sum_{k=1}^{N} (u_{ik})^m x_k}{\sum_{k=1}^{N} (u_{ik})^m} \tag{8}$$

in which $i = \overline{1,C}$.

After obtaining v_i^R, v_i^L, type-reduction is applied to get centroid of clusters as follows:

$$v_i = (v_i^R + v_i^L)/2 \tag{9}$$

For membership grades:

$$u_i(x_k) = (u_i^R(x_k) + u_i^L(x_k))/2, j = 1, ..., C \tag{10}$$

in which

$$u_i^L = \sum_{l=1}^{M} u_{il}/M, u_{il} = \begin{cases} \overline{u}_i(x_k) \text{ if } x_{il} \text{ uses } \overline{u}_i(x_k) \text{ for } v_i^L \\ \underline{u}_i(x_k) \qquad otherwise \end{cases} \tag{11}$$

$$u_i^R = \sum_{l=1}^{M} u_{il}/M, u_{il} = \begin{cases} \overline{u}_i(x_k) \text{ if } x_{il} \text{ uses } \overline{u}_i(x_k) \text{ for } v_i^R \\ \underline{u}_i(x_k) \qquad otherwise \end{cases} \tag{12}$$

Next, defuzzification for IT2FCM is made as if $u_i(x_k) > u_j(x_k)$ for $j = 1, ..., C$ and $i \neq j$ then x_k is assigned to cluster i.

2.3 Genetic Algorithm

The GA [20] is an artificial system based on the principle of natural selection. As a stochastic algorithm, GA is a robust and powerful optimization method for solving problems with a large search space which are not easily solved by exhaustive methods. Usually, a basic GA consists of three operators: selection, crossover, and mutation [19].

3 Genetic-Based Interval Type-2 FCM Clustering

3.1 Chromosome Representation

In GA applications, the unknown parameters are encoded in the form of strings, so-called chromosomes. A chromosome is encoded with binary, integer or real numbers. In this research a chromosome is encoded with a unit which represents a potential cluster centroid. The length of the chromosome, K, is equivalent to the number of clusters in the classification problem. K takes value in the range $[K_{min}, K_{max}]$, where K_{min} is usually assigned to 2 and K_{max} describes the maximum chromosome length, which means the maximum number of possible cluster centroids. Therefore, K_{max} must be selected according to experience. Without assigning the number of clusters in advance, a variable string length is used. Invalid (non-existing) clusters are represented with negative integer ”-1”. The values of the chromosomes are changed in an iterative process to determine the correct number of clusters (the number of valid units in the chromosomes) and the actual cluster centroids for a given classification problem.

3.2 Population Initialization

In genetic algorithm, the population size of P is needed. In the proposed method, all values are chosen randomly from the data space. Such a chromosome belongs to the so-called parent generation. One (arbitrary)chromosomes of the parent generation is of size K. Each chromosome of the population is a potential solution by IT2FCM algorithm with number of clusters $C = K$.

3.3 Selection

This fitness level is used to associate a probability of selection with each individual chromosome. Roulette wheel selection is applied, a proportional selection algorithm where the number of copies of a chromosome, that go into the mating pool for subsequent operations, is proportional to its fitness. If f_i is the fitness of individual P_i in the population, its probability of being selected is as follows:

$$p_i = \frac{f_i}{\sum\limits_{i=1}^{P} f_i} \tag{13}$$

where P is the number of individuals in the population.

3.4 Crossover

The purpose of the crossover operation is to create two new individual chromosomes from two existing chromosomes selected randomly from the current population. Typical crossover operations are one-point crossover, two-point crossover, cycle crossover and uniform crossover. In this research, only the simplest one, the one-point crossover with a fixed crossover probability of μ_c is used; For the one-point crossover, two chromosomes are randomly chosen from the population. Assuming the length of the chromosome to be k, this process randomly chooses a point between 1 and $k-1$ and swaps the content of the two chromosomes beyond the crossover point to obtain the offspring. A crossover between a pair of chromosomes is affected only if they satisfy the crossover probability.

3.5 Mutation

During mutation, all the chromosomes in the population are checked unit by unit and according to a fixed probability μ_m. All values of a specific unit may be randomly changed. In this paper, a number σ in the range $[0,1]$ is generated with uniform distribution. If the value at a gene position is v, after mutation it becomes

$$\begin{aligned} v = v + \sigma * v, if \quad v > 0 \\ v = v + \sigma, if \quad v = 0 \end{aligned}$$

3.6 Validity Index and Fitness Computation

The cluster validity measure used in this paper is the one proposed by Ramze Rezaee [10]. It aims at minimizing the validity index given by the function

$$V_{CWB} = \alpha Scat(C) + Dis(C) \tag{14}$$

The term $Scat(C)$ of V_{CWB} is the average of scattering within clusters, which is defined as

$$Scat(C) = \frac{\frac{1}{C}\sum_{i=1}^{C} ||\sigma(v_i)||}{||\sigma(X)||} \tag{15}$$

in which $||X|| = (X^T.X)^{1/2}$ and

$$\sigma(v_i) = \frac{1}{n}\sum_{k=1}^{n} u_{ik}(x_k - v_i)^2 \tag{16}$$

$$\sigma(X) = \frac{1}{n}\sum_{k=1}^{n}(x_k - \bar{x})^2 \tag{17}$$

with $\bar{x} = \sum\limits_{k=1}^{n} x_k/n$ and $x_k \in X$.

The $Scat(C)$ term is used to measure the compactness of the clusters. The $Dis(C)$ term is the total separation between the clusters which is defined as follows:

$$Dis(C) = \frac{D_{max}}{D_{min}}\sum_{k=1}^{C}(\sum_{z=1}^{C} \parallel v_k - v_z \parallel)^{-1} \tag{18}$$

where $D_{max} = maximum(\parallel v_i - v_j \parallel)$ and $D_{min} = minimum(\parallel v_i - v_j \parallel), \forall i,j \in 1,...,C$. Lastly, α is a weighting factor, given as

$$\alpha = Dis(C_{max}) \tag{19}$$

From validity index, fitness computation is implemented in three following steps.

Step 1: The pixel dataset is clustered according to the centroid encoded in the considered chromosome, such that each pattern $x_i, i = \overline{1,N}$ is assigned to cluster with centroid $v_j, j = \overline{1,K}$ according to the equations (6), (7), (10) in IT2FCM.

Step 2: This step adjusts the values of centroids encoded in the chromosome and replaces them by the mean points of clusters, respectively. The new center v_i^* for the cluster C_i is given by iterative algorithm for finding centroids (9) in IT2FCM.

$$v_i^* = \frac{v_i^R + v_i^L}{2} \tag{20}$$

Step 3: Validity index is computed as above description to obtain V_{CWB}. The goal for achieving a proper clustering is to minimize the V_{CWB}. Thus, the fitness function for chromosome j is defined as $1/V_{CWB}^j$, which is equivalent to the cluster with the smallest inner-cluster scatter and the largest cluster separation.

Therefore, the fitness function is defined as

$$f = \frac{1}{V_{CWB}} \tag{21}$$

3.7 Algorithm

Performance this algorithm is given by a sequence of steps, which are:

1. Generate initial population.

REPEAT

2. Evaluate population followed fitness function by IT2FCM. In this step, We do an IT2FCM step to cluster the processing data. After IT2FCM step, we can calculate the fitness function of the chromosomes following the subsection Validity index and fitness computation 21.

3. Selection.

4. Crossover.

5. Mutation.

6. Reinsertion of new individuals to the population.

UNTIL Termination criterion is met

Termination criterion: We execute the processes of fitness computation, selection, crossover, and mutation for a predetermined number of iterations or the difference between these two fitness values lies below a pre-defined threshold. In every generational cycle, the fittest chromosome till the last generation is preserved. Thus on termination, this chromosome gives us the best solution encountered during the search.

4 Experiments

Experiments are implemented with the following parameters of the GIT2FCM are set: The size of population, P, is taken 30, selection is roulette wheel, crossover rate, $\mu_c = 0.9$ and mutation rate, $\mu_m = 0.01$ [14]. Over the experimental results, the algorithm uses the terminating condition with the number of iterations is set to 20 or the difference between these two fitness values (error) is smaller than 0.00001. Because the experiments are well-known with the number of clusters smaller than 10, we choose the value of chromosome length K is 11.

The results of the proposed algorithm will be compared with the results of the FCM, IT2FCM with the number of the clusters is optimal obtained by GIT2FCM and the initial centroids are randomly selected. Besides, the results of GFCM with the similar combination between GA and FCM with the same parameters such as $P = 30$, $\mu_c = 0.9$, $\mu_m = 0.01$, $K = 11$ also are compared.

In addition, we measured results on the basis of several validity indexes to assess performance of the algorithms.

The first experiment is done with image segmentation on the Wolf Image by GFCM algorithm and GIT2FCM (the proposed algorithm). In Fig. 1, we computed the validity index [10] on this image with different number of clusters. We can see that the number of clusters is 4 in which has the minimum validity index with both GFCM and GIT2FCM. However, the value of validity index of GIT2FCM is smaller than the one of GFCM on the same number of clusters. Fig. 4 shows that image segmentation by GIT2FCM gives the better clustering by separating three clusters clearer involving the wolf, the grass and background.

In the second experiment, the well-known datasets consisting IRIS, Wisconsin Diagnostic Breast Cancer (WDBC), Wine [25] are considered. Implementation is performed to find the optimal number of clusters for these data by two algorithms: GFCM and GIT2FCM (the proposed algorithm) with validity index V_{CWB}. The optimal number of clusters C_{opt} is input of FCM and IT2FCM to cluster and compute validity indices. The considered validity indices consist the Bezdeks partition coefficient (PC-I), the Dunns

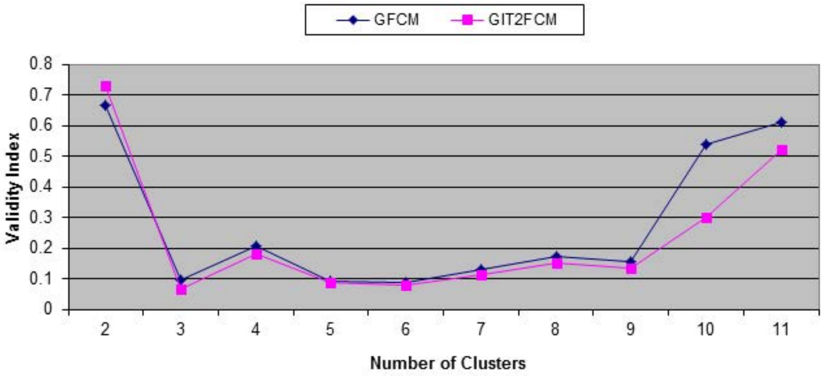
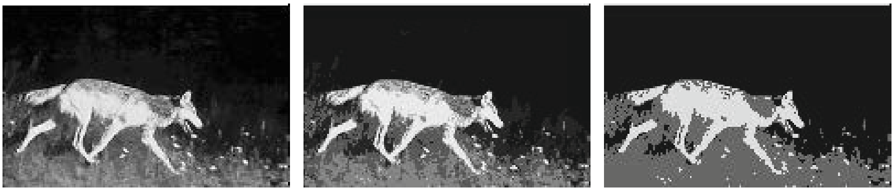


Fig. 1. Validity index is achieved by GFCM and GIT2FCM on Wolf Image



(a) Wolf image (b) GFCM result (c) GIT2FCM result

Fig. 2. Clustering results of GFCM with the number of clusters is 6 and the validity index is 0.085446; GIT2FCM with the number of clusters is 3 and the validity Index is 0.064709364

Table 1. The validity indices V_{CWB} on various algorithms. The optimal clusters C obtained is 3 for Iris data and Wine data, $C = 2$ for WDBC.

Data + Algorithm	2	3	4	5	6	7	8	9	10	11
Iris,GIT2FCM	0.2424	**0.2044**	0.2669	0.2831	0.3066	0.3546	0.2202	0.4893	0.3982	0.2973
Iris,GFCM	1.6326	**1.4228**	2.8368	2.4591	1.6975	2.3160	4.1007	4.1276	3.9259	2.2807
WDBC,GIT2FCM	**0.0015**	0.0018	0.0017	0.0018	0.0020	0.0019	0.0020	0.0021	0.0021	0.0023
WDBC,GFCM	**0.0020**	0.0045	0.0074	0.0075	0.0114	0.0024	0.0075	0.0027	0.0029	0.0205
Wine,GIT2FCM	0.0045	**0.0035**	0.0036	0.0047	0.0048	0.0051	0.0054	0.0058	0.0059	0.0059
Wine,GFCM	0.0099	**0.0043**	0.0095	0.0087	0.0198	0.0172	0.0101	0.0254	0.0166	0.0231

separation index (Dunn-I), the Davies-Bouldins index (DB-I), and the Separation index (S-I), Xie and Beni's index (XB-I), Classification Entropy index (CE-I) [22], Turi's index (T-I) [17], Yuangang Tang's Index (YT-I) [21], CBW index (CBW-I). Results on validity indices are shown in the table 1.

Table 2. The various validity indices on WDBC, Wine and IRIS data by algorithms (1): FCM, (2): GFCM, (3): IT2FCM and (4): GIT2FCM

Validity index	WDBC data ($C = 2$)				Wine data ($C = 3$)				IRIS data ($C = 3$)			
	(1)	(2)	(3)	(4)	(1)	(2)	(3)	(4)	(1)	(2)	(3)	(4)
T-I	1.5713	0.8231	0.5131	0.4481	0.7721	0.3986	0.2356	0.1468	0.6423	0.3667	0.3582	0.0970
DB-I	4.1590	4.1590	1.2750	1.0359	5.6337	4.7694	0.7972	0.5326	4.9394	2.9907	1.3350	1.2805
XB-I	0.0103	0.0060	0.0026	0.0021	0.0159	0.0116	0.0067	0.0060	0.0137	0.0115	0.0113	0.0040
S-I	0.0597	0.0597	0.0024	0.0001	0.0441	0.0421	0.1732	0.0008	0.0374	0.0349	0.0195	0.0006
CE-I	0.1809	0.1809	0.1722	0.0612	0.3804	0.3733	0.1943	0.0815	0.3986	0.2211	0.0943	0.0909
PC-I	0.8969	0.8969	0.9059	0.9394	0.7948	0.7993	0.8083	0.9265	0.7832	0.7876	0.9516	0.9897
YT-I	8.7454	8.7453	7.8837	1.0003	7.7540	7.4410	6.3607	2.0046	5.9636	5.5663	4.7094	1.6805
CWB-I	0.0020	0.0020	0.0017	0.0015	0.0099	0.0044	0.0081	0.0035	1.5687	0.7467	0.7073	0.2635

Various validity indices are calculated with the optimal number of clusters that are shown in the tables 2.

Because, the validity indices are proposed to evaluate the quality of clustering. The better algorithm gives the smaller value of T-I, DB-I, XB-I, S-I, CE-I, YT-I ,CWB-I and the larger value of PC-I. The summarized results show that the GIT2FCM clustering (the proposed algorithm) have a better performance or higher quality clustering than the other typical algorithm such as FCM, IT2FCM and GFCM. Besides, GIT2FCM can automatically obtains the optimal number of clusters.

5 Conclusion

One of the priori inputs traditionally needed for unsupervised classification is the number of clusters in the data set. In many cases, however, this number of classes is not available. This paper presents a clustering algorithm based on genetic technique which determines the required number of clusters as part of the algorithm. The proposed approach do not need to predict the optimal number of clusters, required to partition the dataset. The experiments are done based on well-known dataset with the statistics show that the algorithm generates clusters with better quality.

The next goal is some studies related to speed-up algorithms for processing huge data based parallel architecture of GPU computing and applying the algorithm to problems.

References

1. Hwang, C., Rhee, F.C.H.: Uncertain Fuzzy clustering: Interval Type-2 Fuzzy Approach to C-Means. IEEE Tran. on Fuzzy Systems 15, 107–120 (2007)
2. Kanungo, T., Mount, D.M., Netanyahu, N.S., Piatko, C.D., Silverman, R., Wu, A.Y.: An Efficient k-Means Clustering Algorithm: Analysis and Implementation. IEEE Trans. on Pattern Analysis and Machine Intelligence 24(7), 881–893 (2002)

3. Hung, M.C., Wu, J., Chang, J.H., Yang, D.L.: An Efficient k-Means Clustering Algorithm Using Simple Partitioning. J. of Info. Science and Engineering 21, 1157–1177 (2005)
4. Zalik, K.R.: An efficient k-means clustering algorithm. Pattern Recognition Letters 29, 1385–1391 (2008)
5. Abdul Nazeer, K.A., Sebastian, M.P.: Improving the Accuracy and Efficiency of the k-means Clustering Algorithm. In: Proceedings of the World Congress on Engineering (2009)
6. Mendel, J.M., John, R.I.: Type-2 fuzzy set made simple. IEEE Trans.Fuzzy Syst. 10(2), 117–127 (2002)
7. Karnik, N., Mendel, J.M.: Operations on Type-2 Fuzzy Sets. Fuzzy Sets and Systems 122, 327–348 (2001)
8. Liang, Q., Mendel, J.M.: Interval Type-2 Fuzzy Logic Systems: Theory and Design. IEEE Trans. on Fuzzy Systems 8(5), 535–550 (2000)
9. Mendel, J.M., John, R.I., Liu, F.: Interval Type-2 Fuzzy Logic Systems Made Simple. IEEE Trans. on Fuzzy Systems 14(6), 808–821 (2006)
10. Rezaee, M.R., Lelieveldt, B.P.F., Reiber, J.H.C.: A new cluster validity index for the fuzzy c-mean. Pattern Recognition Lett. 19, 237–246 (1998)
11. Bezdek, J.: Pattern Recognition with Fuzzy Objective Function Algorithms. Plenum, New York (1981)
12. Alata, M., Molhim, M., Ramini, A.: The Optimizing of Fuzzy C-Means Clustering Algorithm Using GA. World Academy of Science, Engineering and Technology 2, 224–229 (2008)
13. Abraham, A., et al.: Foundations of Computer Intelligent. SCI, vol. 4, 204, pp. 167–195 (2009)
14. Srinivas, M., Patnaik, L.M.: Genetic Algorithms: A Survey. IEEE Computer 27(6), 17–26 (1994)
15. Steinley, D., Brusco, M.J.: Initialization k-means batch clustering: a critical evaluation of several techniques. J. Classif. 24, 99–121 (2007)
16. Cheug, Y.M.: On rival penalization controlled competitive learning for clustering with automatic cluster number selection. IEEE Trans. Knowledge Data Eng. 17, 1583–1588 (2005)
17. Roberts, S.J., Everson, R., Rezek, I.: Maximum certainty data partitioning. Pattern Recognition 33, 833–839 (2000)
18. Turi, R.H.: Clustering-Based Color Image Segmentation, PhD Thesis, Monash University, Australia (2001)
19. Haykin, S.: Neural Networks: A Comprehensive Foundation. Prentice-Hall, Inc. (1999)
20. Holland, J.H.: Adaptation in Natural and Artificial Systems. MIT Press (1975)
21. Tang, Y., Sun, F.: Improved Validation Index for Fuzzy Clustering. In: American Control Conference (2005)
22. Wang, W., Zhang, Y.: On fuzzy cluster validity indices. Fuzzy Sets and Systems 158, 2095–2117 (2007)
23. Cao, H., Deng, H., Wang, Y.: Segmentation of M-FISH Images for Improved Classification of Chromosomes With an Adaptive Fuzzy C-means Clustering Algorithm. IEEE Transactions on Fuzzy Systems 20(1), 1–8 (2012)
24. Wei, C., Tingjin, L., Jizheng, W., Yanqing, Z.: An improved genetic FCM clustering algorithm, An improved genetic FCM clustering algorithm. In: International Conference on Future Computer and Communication, ICFCC, pp.V1-45–V1-48 (2010)
25. Datasets, `http://www.ics.uci.edu/mlearn/mlrepository.html`