Journal of King Saud University – Computer and Information Sciences (2018) 30, 25–32

King Saud University

Journal of King Saud University – Computer and Information Sciences

www.ksu.edu.sa
www.sciencedirect.com

Fuzzy clustering based on Forest optimization algorithm

Arash Chaghari *, **Mohammad-Reza Feizi-Derakhshi, Mohammad-Ali Balafar**

Department of Electrical and Computer Engineering, University of Tabriz, Tabriz, Iran

Received 12 June 2016; revised 6 August 2016; accepted 20 September 2016
Available online 28 September 2016

KEYWORDS

Fuzzy clustering;
Partition matrix;
Forest optimization;
Gradient method;
Clustering index

Abstract Clustering is one of the classification methods for data analysis and it is one of the ways of data analysis, too. There are various methods for fuzzy clustering using optimization algorithms such as genetic algorithm and particle swarm optimization algorithm that were specified. In this paper, the combination of one of the recent optimization algorithms called Forest optimization algorithm and one of the local search methods called gradient method are used to perform fuzzy clustering. The purpose of applying the gradient method is accelerating the convergence of the used optimization algorithm. To apply the proposed method, 4 types of real data sets are used. Cluster validity measures are used to obtain and verify the accuracy of the proposed method (FOFCM). By analyzing and comparing the results of the proposed method with the results of algorithms GGAFCM (fuzzy clustering based on genetic algorithm) and PSOFCM (fuzzy clustering based on particle swarm optimization algorithm), it has been shown that the accuracy of the proposed approach is significantly increased.

1. Introduction

Clustering is a classification way for data analysis, which is utilized to classify a set of data or patterns commonly multidimensional into different groups according to a predefined measure, in order that items in the same group are more almost the same than those in different groups. All the more particularly, the patterns that are generally s dimensional vectors are conveyed to c classes while certain sort of optimization criterion is minimized, and the patterns in the same class are more comparable than those in various classes at last. In recent decades, clustering plays the key role in different fields of science and engineering, such as data analysis, pattern recognition, machine learning, image segmentation, error detection and so on.

In general, clustering methods are divided into two general categories; crisp and fuzzy. The degree of the membership of each sample of the data is zero or one in crisp methods. In fact, crisp methods can be considered as a special case of fuzzy algorithms. In other words, the membership value of the sample that belongs to a cluster is one and its membership value for the rest of the clusters is zero. The advantage of crisp methods

* Corresponding author.
E-mail addresses: a.chaghari@tabrizu.ac.ir (A. Chaghari), mfeizi@tabrizu.ac.ir (M.-R. Feizi-Derakhshi), balafarila@tabrizu.ac.ir (M.-A. Balafar).
Peer review under responsibility of King Saud University.

Production and hosting by Elsevier

http://dx.doi.org/10.1016/j.jksuci.2016.09.005
1319-1578

is its easiness and efficiency to implement. One of the famous algorithms in this area is the algorithm k-means (Forgy, 1965). Although these algorithms are widely used and have developed well, they are not appropriate for fuzzy data set. For this category of algorithms, it is assumed that the data set classes have nothing in common to one another and are completely separated from each other. On the other side of crisp methods, membership degree of the samples is set in the interval [0, 1] in fuzzy methods.

Bezdek developed a fuzzy clustering algorithm, the well-known fuzzy c-means (FCM) (Bezdek, 1973a). The algorithm is the fuzzy equivalence of the algorithm k-means. According to FCM usage, a lot of algorithms are presented to improve the accuracy of clustering. In the standard FCM algorithm and all the proposed methods for its improvement, the number of clusters should have already been set. In other words, in such circumstances, clustering problem can be defined as follows: n sample with s dimension should be in c cluster, so that each sample should be alleged in the corresponding cluster. So, there is an evaluation function that the cluster result is evaluated by and its purpose is to optimize the evaluation function by which, an optimal clustering is achieved.

Global optimization algorithms known as genetic algorithms (Bezdek and Hathaway, 1994; Maulik and Bandyopadhyay, 2000; Bandyopadhyay and Maulik, 2001), ant colony optimization (Dorigo et al., 1996), particle swarm optimization (Liu et al., 2005; De Falco et al., 2007) and chaos optimization (Li et al., 2008) are well-known algorithms to optimize fuzzy clustering. In other words, several researchers formulated the entire clustering task of FCM explicitly as an optimization problem and solved it using various metaheuristics viz., simulated annealing (Granelli et al., 1989; Victoire and Jeyakumar, 2005), variable neighborhood search (Li et al., 1997), genetic algorithms (Han et al., 2001; Victoire and Jeyakumar, 2005), tabu search (Walters and Sheble, 1993) and threshold accepting (Panigrahi et al., 2006) were suggested. Recently, Jayabarathi et al. (2005) applied DE after FC so that it can lead to a global optimum. DE was also used with FCM in several different ways. Gaing (2003) presented a real-coded modified DE based automatic fuzzy clustering algorithm which automatically evolves the number of clusters as well as the proper partitioning from a data set. Passino (2002) proposed an evolutionary-fuzzy clustering algorithm for automatically grouping the pixels of an image into different homogeneous regions. An improved variant of the DE was used to determine the number of naturally occurring clusters in the image as well as to refine the cluster centers. Mishra (2005) used DE to optimize the coordinates of the samples distributed randomly on a plane.

Researchers have tried to improve FCM by introducing excellent optimization methods to optimize the objective function of FCM, trying to avoid trapping into local minima. In Karaboga and Ozturk (2010), bee colony optimization algorithm is used and combined with the algorithm FCM, to cluster data. In the algorithm (Xiaoqiang and Jinhu, 2014) a combination of invasive weed optimization algorithm and clustering algorithm FCM is used so that clustering of the data is done. In the algorithm CPSFC (Li et al., 2012), a combination of particle swarm optimization algorithm, chaotic local search, and gradient method is used to provide good performance in capturing the global optimal fitness, thus getting the best clustering results.

The paper is organized as follows: in Section 2, basic concepts including standard FCM algorithm, Forest optimization algorithm, gradient method and validity indices of fuzzy clustering are mentioned. Section 3 describes the proposed method and in Section 4 the results of the implementation of the proposed method on the data set are shown. In Section 5, conclusions and future work are mentioned.

2. Basic concepts

In this section, the algorithm FCM, Forest optimization algorithm, and gradient method will be discussed. The noted meanings are prerequisite toward the proposed method. Also, for the proposed method evaluation, the evaluation measures will be described.

2.1. FCM algorithm

The main part of fuzzy clustering, is to determine similarity measure by which the distance between the patterns can be determined. In the algorithm FCM, the Euclidean distance is used as similarity measure. Fitness function that is used in FCM algorithm is defined as:

$$J_m = \sum_{i=1}^{c}\sum_{j=1}^{n}(u_{i,j})^m \|y_j - z_i\|_A^2 \tag{1}$$

where $Y = (y_1, y_2, \ldots, y_n)$ is the data set that the number of features or dimensions of each sample is equal to s. $Z = (z_1, z_2, \ldots, z_c)$ is the center of clusters. $U = [u_{i,j}]_{c\times n}$ is the partition matrix, $U_{ij} \in [0, 1]$ is interpreted to be the grade of membership of x_j in the ith cluster. Symbol $\|\cdot\|_A$ means norm of matrix A. If A equals the identity matrix, the phrase $\|y_j - z_i\|$ means the Euclidean distance from y_j to the ith cluster center. It is believed the minimization of J_m will produce the best cluster structure and the optimal cluster results.

The minimization of J_m can be reached by Lagrange multiplier method while the partition matrix U and cluster centers Z have expressions as follows:

$$\boldsymbol{u_{i,j}} = \left[\sum_{k=1}^{c}\left(\frac{d_{i,j}}{d_{k,j}}\right)^{\frac{2}{m-1}}\right]^{-1} \boldsymbol{1 \leqslant i \leqslant c;\ 1 \leqslant j \leqslant n} \tag{2}$$

$$z_i = \frac{\sum_{j=1}^{n}(u_{i,j})^m y_j}{\sum_{j=1}^{n}(u_{i,j})^m} \tag{3}$$

By repeating Eqs. (2) and (3), the fitness function J_m tends toward its minimum value gradually. The algorithm FCM can be expressed as follows:

1. Set the cluster numbers c, set initial cluster centers $z_i^{(0)}$, $1 \leqslant i \leqslant c$, and set the tolerance ε to determine when to stop the algorithm.
2. Acquiring new values of u and z using Eqs. (2) and (3).
3. Calculating the value of the difference between the new cluster centers and the new degree of membership of the second phase of their previous values. If earned value is less than the threshold error ε or the number of iteration is

equal to the maximum value, the algorithm will be terminated; otherwise, the second step is performed.

The FCM algorithm can be considered as a kind of local search. So, being located in local minimum and being sensitive to initial cluster centers, are the main problems of FCM algorithm, so that different values for the initial cluster centers, produces different results.

2.2. Forest optimization algorithm

The Forest optimization algorithm (FOA) (Ghaemi and Feizi-Derakhshi, 2014) is suitable for continuous non-linear optimization problems. The algorithm is inspired by the existence of ancient trees after decades. While many of the trees are short-lived, the number of trees is still in existence even after a few decades. In this algorithm, spreading seeds of the trees are simulated so that the number of seeds is set under the trees, and the number of seeds spread in a vast area of Forest by natural events such as wind. The output of the algorithm suggests improving of its accuracy in finding the optimal positions rather than genetic algorithm and particle swarm optimization.

This algorithm has three main steps: (1) local seed production, (2) removing some members of the population, (3) global seed production. In this algorithm, like other evolutionary algorithms, forest trees are initialized in the initialization step. A tree, in addition to the values of the variables, has a part that represents the age of the related tree. The age of each tree is zero at first. After initialization of the trees, in local seed production step, some new trees aged zero (new seeds) are created. Then, one unit is added to the age of previous trees. In the second step, number of trees based on the number of pre-defined population, should be removed from the population.

Removing the excess trees is based on their fitness function values. Therefor some trees will be omitted from the forest and they will form the candidate population for global seeding step. In the production of global seeds, a percentage of the candidate population is chosen to move far in the forest. Global seeding step adds some new potential solutions to the forest so that go away from local optimums. Finally, after sorting the trees according to their fitness value, the tree with the highest fitness value is selected as the best tree. Then the age of best tree will be set to 0 in order to avoid the aging of the best tree as the result of local seeding step.

2.3. Gradient method

Gradient method is an improved local search algorithm that its procedures are as follows:

1. Set the number of iteration and specify the cluster centers Z, from the obtained optimal solution of optimization algorithm.
2. Repeat steps 3–5 to reach the stop condition.
3. Get the partition matrix U from the cluster centers Z.
4. Update the cluster centers Z using the partition matrix U.
5. Stop the algorithm, if it is reached the maximum number of iteration.

2.4. Cluster validity measures

Cluster validity indices are used for evaluating the output of fuzzy clustering algorithms. In other words, this type of indices is for evaluating the relative well-being clusters created on data set. The clustering algorithm output, regardless of being fuzzy or crisp, is the partition matrix U and the cluster centers Z. Well-being of created clusters is evaluated regarding matrix U and matrix Z.

There are many cluster validity indices. In this article, results are expressed in terms of indices PC (Bezdek, 1973b), PE (Bezdek, 1975) and XB (Xie and Beni, 1991). Values of the three indices are obtained in Eqs. (4)–(6).

$$\mathrm{PC} = \frac{1}{n}\sum_{i=1}^{c}\sum_{j=1}^{n}u_{i,j}^{2} \tag{4}$$

$$\mathrm{PE} = -\frac{1}{n}\sum_{i=1}^{c}\sum_{j=1}^{n}u_{i,j}\log_b(u_{i,j}) \tag{5}$$

$$\mathrm{XB} = \frac{\sum_{i=1}^{c}\sum_{j=1}^{n}u_{i,j}\|y_j - z_i\|_A^2}{n \times min_{i\neq j}\|z_i - z_j\|^2} \tag{6}$$

The property of this index has been studied in Bezdek (1973b), and it is believed PC gets its maximum value when the cluster structure is optimal. PE takes its minimum value when the cluster structure is optimal. XB index reaches its minimum value when the partition is the best.

3. Proposed method

In this paper, we have tried to improve the accuracy of the standard FCM algorithm using one of the optimization algorithms called the Forest optimization algorithm (Ghaemi and Feizi-Derakhshi, 2014). The proposed method (FOFCM) starts fuzzy clustering using a combination of the Forest optimization algorithm with the gradient method. Using the Forest optimization algorithm, optimized cluster centers can be obtained. To do this, the evaluation function of the standard FCM algorithm (Eq. (1)), is used as fitness function for the Forest algorithm. The solution may be equivalent to a tree in the forest. To do optimization, cluster centers are considered as optimization variables. In other words, the solution may include all cluster centers. So if c, is the number of clusters and each cluster centers, has s features (dimensions), the length of each optimization variables will be equal to $c \times s$. The vector show of the solution of X_i, is as $X_i = [x_{i,1}, x_{i,2}, \ldots, x_{i,s}, x_{i,c\times(s-1)+1}, x_{i,c\times(s-1)+2}, \ldots, x_{i,c\times s}]$. The first s number of vector X_i is first cluster center and second s number is the second cluster center and so on. The gradient method is used in order to speed up the convergence of the algorithm and to provide good performance in capturing the global optimal fitness, thus getting the best clustering result. In each iteration of the Forest, the fitness function for the best obtained result is compared with the output of gradient method. If the fitness function value for the output of gradient method is better than the fitness function value of the optimal solution derived from the Forest algorithm that will be substituted. When the Forest algorithm terminates, the best obtained solution is formed in the form of cluster centers and the partition matrix is obtained by the use of cluster centers. Finally, the accuracy of suggested algorithm is obtained using cluster validity indices.

Details of the implementation of the proposed method are as follows:

1. Initialization: in this step, the initial set of trees, are randomly initialized. Each of trees is a selected solution for cluster centers. In each solution, the age of each tree (solution) and the value of the fitness function (J_m) are stored, in addition to the cluster centers. At first, the age of each solution (or tree) is equal to 0. Since the obtained optimal solution is compared with the optimal solution of the previous at the end of each iteration of the Forest optimization algorithm, in the first iteration, the optimal solution is selected randomly.
2. Local seeding of the trees: in this step, based on the value of "Local Seeding Changes" or "LSC", new solutions for each of the available solutions with 0 age, are created. In fact, at this step, new solutions are examined around existing solutions (exploitation).
3. Removing some of the solutions: at this step, regarding the allowed given number of the solutions in the population, and also allowed age of the solutions in the population, the number of the solutions will be deleted from the main population and are added to the population called the candidate population. Removing the extra solutions, the solutions with the age that is more than the maximum allowed age of a tree (life time), are deleted at first. Then, if the number of members of the population is more than the maximum number of population, solutions that the value of their fitness function is worse than the rest of the population, will be omitted (elitism selection). The deleted solutions are added to the candidate population.
4. Global seeding of the trees: in this step, the trees according to "transfer rate" from the candidate population are selected. To do this, regarding the transfer rate, a predefined percentage of the population is chosen for new solutions at first. Then, the number of variables that should be changed is determined based on the parameter value "Global Seeding Changes" or "GSC". After the random selection of the variables (genes) of a solution, their value is assigned to random values (mutation). Then, the seeds or new solutions are added to the main population. This step is to avoid being trapped in local optimum (exploration).
5. Repeating the step of removing some of the solutions: since the new number of solutions was added to the main population in the fourth step, the number of solutions of the population is selected as the optimal ones and rest of the solutions are removed from the main population based on the fitness function.
6. Choosing the optimal solution: in this step, the optimal solution of the end population is selected based on the value of the fitness function. If the obtained optimal solution is better than the optimal solution of the previous iteration of the algorithm, it will be replaced.
7. Running the gradient method: The gradient method is run in this step and its output is compared with the output of step 6. If the output value of the gradient method is better than the result of the 6 step, it will be replaced.

When the Forest algorithm terminates, the best obtained solution will be formed as cluster centers and partition matrix will be obtained by the use of these centers. At the standard FCM algorithm, cluster centers are obtained using the partition matrix at the first iteration of the algorithm. The partition matrix is also set randomly by the first iteration. Therefore, the cluster centers are set randomly in the first iteration of standard FCM algorithm. In this paper, FCM algorithm is not used in the standard form. That is in the third step, the optimal cluster centers which are obtained from the second step, are set as an input to the algorithm FCM. So the value of the cluster centers at the first iteration of algorithm FCM is equal to the final result of the Forest optimization algorithm and the gradient method.

4. Experimental results

Due to that fact that at optimization algorithms, gained solutions at each time of algorithm running, are not constant values, each of the compared algorithms in this section is run 50 times independently and the mean and standard deviation of 50 times of running has been shown. For the implementation of the proposed method, 4 datasets named iris, wine, glass, and liver disorder have been used. As we see in Table 1, the information of data sets is as follows:

In this section, real data sets are applied to test the performance of FOFCM algorithm. Four real data sets are used in experiments. The studies focus on the convergence performance and the clustering effect of FOFCM algorithm. In order to make it evident to show the performance of FOFCM, comparative studies are completed, while GGAFCM (Hall et al., 1999) and PSOFCM algorithm (De Falco et al., 2007) are introduced in experiments. Experiments are made up of two aspects, namely, experiments on convergence performance and experiments on clustering effect.

Datasets:

1. Iris: This data set has 155 samples and four features and three clusters (Hall et al., 1999).
2. Wine: Wine is a data set that has 178 samples and 3 features. It contains 6 clusters (Asuncion, 2007).

Table 1 Information of data sets.

Data set	Size	Dimension	Number of clusters
Iris	150	4	3
Wine	178	13	3
Vowel	871	3	6
Liver disorder	345	7	2

Table 2 Parameters for implementation of the Forest optimization algorithm.

# iteration	100
# initial population	50
Life time	15
Transfer rate	10
Local seeding change	20% of # dimensions
Global seeding change	10% of # dimensions

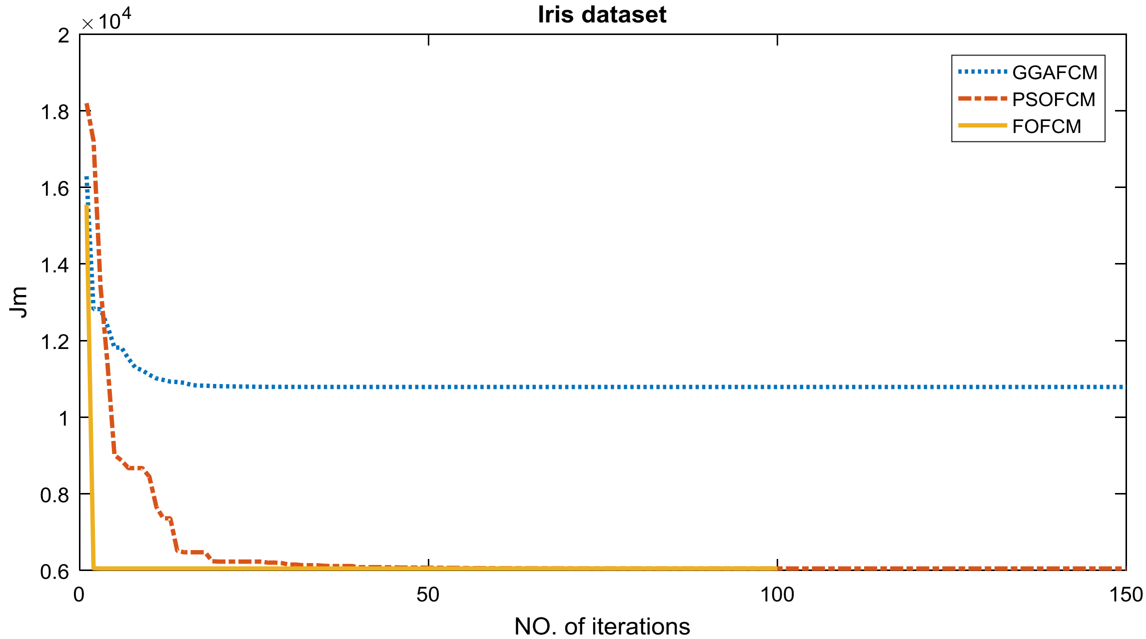

Figure 1 The convergence comparison of different methods on Iris.

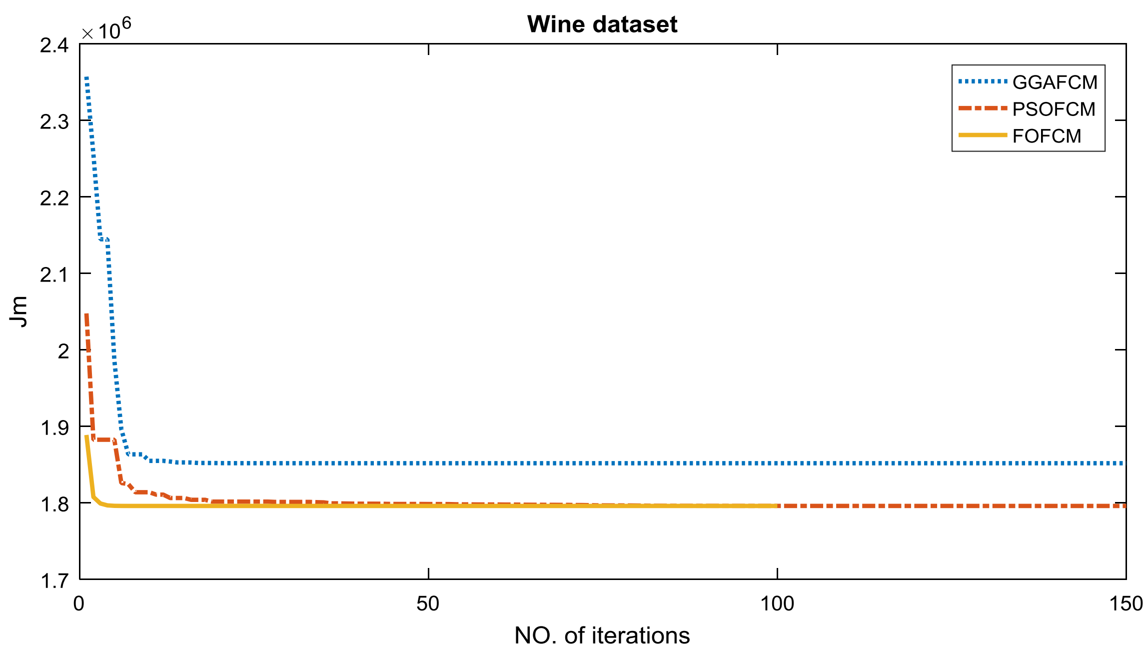

Figure 2 The convergence comparison of different methods on Wine.

Table 3 Comparison of GGAFCM, PSOFCM and FOFCM in terms of optimal values.

Data set	GGAFCM		PSOFCM		FOFCM	
	Mean	Std. dev.	Mean	Std. dev.	Mean	Std. dev.
Iris	8.9911e+03	1.2939e+03	6.0506e+3	1.7278e−9	**6.0506e+03**	**5.0689e−12**
Wine	1.8267e+06	2.9442e+04	1.7961e+06	4.7981	**1.7961e+06**	**7.5921e−10**
Vowel	2.1336e+07	1.2589e+06	1.7108e+07	0.2087	**1.7108e+07**	**2.6055e−08**
Liver disorder	3.8773e+05	4.0718e+04	3.3311e+05	1.0405e−07	**3.3311e+05**	**1.8871e−10**

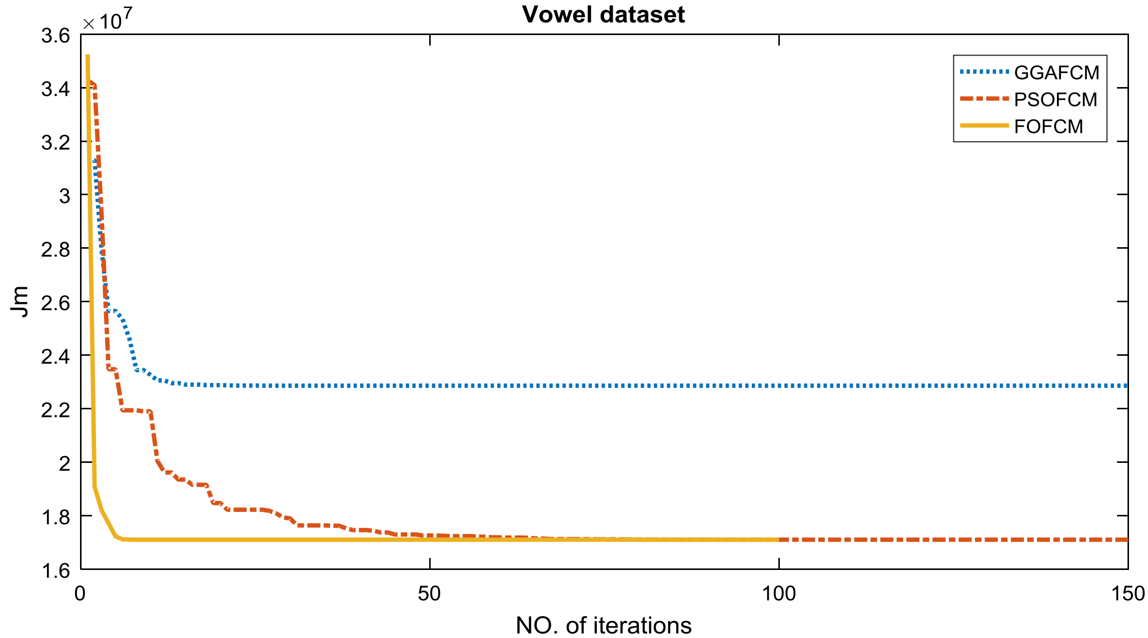

Figure 3 The convergence comparison of different methods on Vowel.

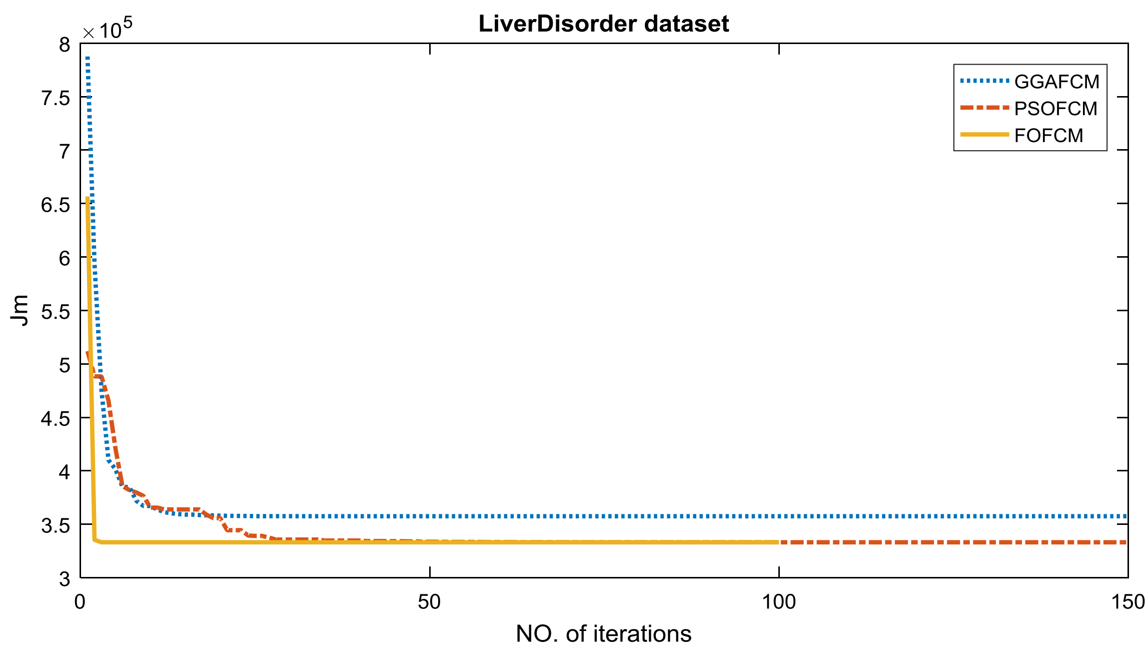

Figure 4 The convergence comparison of different methods on Liver disorder.

Table 4 Comparison of GGAFCM, PSOFCM, and FOFCM in terms of PC validity index.

Data set	GGAFCM		PSOFCM		FOFCM	
	Mean	Std. dev.	Mean	Std. dev.	Mean	Std. dev.
Iris	0.6954	0.0397	0.7834	1.0621e−08	**0.7834**	**2.1879e−10**
Wine	0.7869	0.0047	0.7909	8.1241e−06	**0.7909**	**8.3779e−10**
Vowel	0.4536	0.0276	0.5498	2.6994e−06	**0.5498**	**2.5704e−10**
Liver disorder	0.7911	0.0621	0.8300	3.0806e−08	**0.8300**	**2.4404e−09**

Table 5 Comparison of GGAFCM, PSOFCM, and FOFCM in terms of PE validity index.

Data set	GGAFCM		PSOFCM		FOFCM	
	Mean	Std. dev.	Mean	Std. dev.	Mean	Std. dev.
Iris	0.5393	0.0601	0.3955	1.4741e−08	**0.3955**	**1.5682e−10**
Wine	0.3869	0.0064	0.3804	9.9434e−06	**0.3804**	**9.5282e−10**
Vowel	1.1049	0.0481	0.9224	5.7489e−06	**0.9224**	**2.6535e−09**
Liver disorder	0.3454	0.0836	0.2881	4.5150e−08	**0.2881**	**3.4511e−09**

Table 6 Comparison of GGAFCM, PSOFCM, and FOFCM in terms of XB validity index.

Data set	GGAFCM		PSOFCM		FOFCM	
	Mean	Std. dev.	Mean	Std. dev.	Mean	Std. dev.
Iris	0.4395	0.3319	0.1369	4.8012e−08	**0.1369**	**7.0651e−11**
Wine	0.1272	0.0121	0.1257	3.8199e−05	**0.1257**	**2.2213e−09**
Vowel	0.9403	0.8852	0.1893	1.1671e−05	**0.1893**	**1.2365e−10**
Liver disorder	0.2323	0.2202	0.1261	5.6461e−08	**0.1261**	**2.1498e−09**

Table 7 CPU Time in seconds for different algorithms to clustering the data set.

Data set	GGAFCM	PSOFCM	FOFCM
Iris	3.298045	3.740919	6.769025
Wine	4.283092	4.781740	38.593023
Vowel	10.876156	12.097590	36.745444
Liver disorder	4.336522	4.409214	8.049203

3. Vowel: Vowel is a data set that has 871 samples, 15 features and 6 clusters (Pal and Majumder, 1977).
4. Liver disorder: This data set contains 345 samples that has 7 features and 2 clusters (Asuncion, 2007).

The specific information of real data sets is shown in Table 1. Also, the parameter values for the Forest optimization algorithm according to Ghaemi and Feizi-Derakhshi (2014) are as Table 2. In GGAFCM and PSOFCM algorithms, the number of particles is 50, the maximum inertia weight $w_{max} = 0.9$, the minimum inertia weight $w_{max} = 0.4$, $C_1 = 2$, $C_2 = 2$. The iteration of FOFCM is 100, while PSOFCM iterates 120 times for complete convergence. In GGAFCM algorithm, population of chromosome is 50, and the probability of crossover (Pc) and mutation (Pm) are taken to be 0.85 and 0.008, respectively, and iteration is 100.

Algorithms GGAFCM (Hall et al., 1999) and PSOFCM (De Falco et al., 2007), have been used to be compared with suggested method. Each algorithm has been run 50 times independently. The results of the optimal value of fitness function of each algorithm, is observed in Table 3, where best values are shown in bold. Furthermore, the convergence comparison of different methods on four data sets has been shown in Figs. 1–4.

The number of iterations of gradient method according to Li et al. (2012) is considered as 3. If the number of iterations of gradient method is a lot, optimization algorithm will be caught at a local optimum. As it is observed in Table 3, the proposed method has performed better than GGAFCM algorithms and PSOFCM in obtaining J_m minimum value.

Also the results of PC validity index have been shown in Table 4, the results of PE validity index in Table 5, and the results of XB validity index has been shown in Table 6. Table 7 shows the time per iteration in seconds for different methods on the four real data sets.

5. Conclusion and future work

In this paper, we tried to use an optimization algorithm called Forest algorithm and combine it with a local search optimization method called gradient method, to improve the FCM algorithm. Optimization algorithm may not reach to the optimal value of fitness function of the FCM algorithm in low number of iterations alone. Therefore, a local search method called gradient method was used to increase the speed of convergence of the optimization algorithm. For the FCM algorithm and its improved versions, it is assumed that all the features of the samples in a given data set make equal contribution when constructing the optimal clusters. Some features are more important, and should have more weight. Hence, using the methods of feature weighting, we can increase the importance of some features compared to the rest of the features. And as a result, the accuracy through fuzzy clustering methods can be improved.

References

Asuncion, D.J.N.A., 2007. UCI Machine Learning Repository.

Bandyopadhyay, S., Maulik, U., 2001. Nonparametric genetic clustering: comparison of validity indices. IEEE Trans. Syst. Man Cybern. Part C: Appl. Rev. 31, 120–125.

Bezdek, J.C., 1973a. Fuzzy Mathematics in Pattern Classification.

Bezdek, J.C., 1973b. Cluster Validity with Fuzzy Sets.

Bezdek, J.C., 1975. Mathematical models for systematics and taxonomy. In: Proceedings of Eighth International Conference on Numerical Taxonomy, pp. 143–166.

Bezdek, J.C., Hathaway, R.J., 1994. Optimization of fuzzy clustering criteria using genetic algorithms. In: Proceedings of the First IEEE Conference on Evolutionary Computation, 1994. IEEE World Congress on Computational Intelligence, pp. 589–594.

De Falco, I., Della Cioppa, A., Tarantino, E., 2007. Facing classification problems with Particle Swarm Optimization. Appl. Soft Comput. 7, 652–658.
Dorigo, M., Maniezzo, V., Colorni, A., 1996. Ant system: optimization by a colony of cooperating agents. IEEE Trans. Syst. Man Cybern. Part B Cybern. 26, 29–41.
Forgy, E.W., 1965. Cluster analysis of multivariate data: efficiency versus interpretability of classifications. Biometrics 21, 768–769.
Gaing, Z.-L., 2003. Particle swarm optimization to solving the economic dispatch considering the generator constraints. IEEE Trans. Power Syst. 18, 1187–1195.
Ghaemi, M., Feizi-Derakhshi, M.-R., 2014. Forest optimization algorithm. Expert Syst. Appl. 41, 6676–6687.
Granelli, G., Marannino, P., Montagna, M., Silvestri, A., 1989. Fast and efficient gradient projection algorithm for dynamic generation dispatching. In: IEE Proceedings C-Generation Transmission and Distribution, pp. 295–302.
Hall, L.O., Ozyurt, I.B., Bezdek, J.C., 1999. Clustering with a genetically optimized approach. IEEE Trans. Evol. Comput. 3, 103–112.
Han, X., Gooi, H., Kirschen, D.S., 2001. Dynamic economic dispatch: feasible and optimal solutions. IEEE Trans. Power Syst. 16, 22–28.
Jayabarathi, T., Jayaprakash, K., Jeyakumar, D., Raghunathan, T., 2005. Evolutionary programming techniques for different kinds of economic dispatch problems. Electr. Power Syst. Res. 73, 169–176.
Karaboga, D., Ozturk, C., 2010. Fuzzy clustering with artificial bee colony algorithm. Sci. Res. Essays 5, 1899–1902.
Li, F., Morgan, R., Williams, D., 1997. Hybrid genetic approaches to ramping rate constrained dynamic economic dispatch. Electr. Power Syst. Res. 43, 97–103.
Li, C., Zhou, J., Li, Q., Xiang, X., 2008. A fuzzy cluster algorithm based on mutative scale chaos optimization. In: Advances in Neural Networks-ISNN 2008. Springer, pp. 259–267.
Li, C., Zhou, J., Kou, P., Xiao, J., 2012. A novel chaotic particle swarm optimization based fuzzy clustering algorithm. Neurocomputing 83, 98–109.
Liu, B., Wang, L., Jin, Y.-H., Tang, F., Huang, D.-X., 2005. Improved particle swarm optimization combined with chaos. Chaos, Solitons Fractals 25, 1261–1271.
Maulik, U., Bandyopadhyay, S., 2000. Genetic algorithm-based clustering technique. Pattern Recogn. 33, 1455–1465.
Mishra, S., 2005. A hybrid least square-fuzzy bacterial foraging strategy for harmonic estimation. IEEE Trans. Evol. Comput. 9, 61–73.
Pal, S.K., Majumder, D.D., 1977. Fuzzy sets and decision making approaches in vowel and speaker recognition. IEEE Trans. Syst., Man, Cybern 7, 625–629.
Panigrahi, C., Chattopadhyay, P., Chakrabarti, R., Basu, M., 2006. Simulated annealing technique for dynamic economic dispatch. Electr. Power Compon. Syst. 34, 577–586.
Passino, K.M., 2002. Biomimicry of bacterial foraging for distributed optimization and control. IEEE Control Syst. 22, 52–67.
Victoire, T.A.A., Jeyakumar, A.E., 2005. Reserve constrained dynamic dispatch of units with valve-point effects. IEEE Trans. Power Syst. 20, 1273–1282.
Victoire, T.A.A., Jeyakumar, A.E., 2005. Deterministically guided PSO for dynamic dispatch considering valve-point effect. Electr. Power Syst. Res. 73, 313–322.
Walters, D.C., Sheble, G.B., 1993. Genetic algorithm solution of economic dispatch with valve point loading. IEEE Trans. Power Syst. 8, 1325–1332.
Xiaoqiang, Z., Jinhu, Z., 2014. An improved IWO-FCM data mining algorithm. In: Control and Decision Conference (2014 CCDC), The 26th Chinese, pp. 4997–5001.
Xie, X.L., Beni, G., 1991. A validity measure for fuzzy clustering. IEEE Trans. Pattern Anal. Mach. Intell. 13, 841–847.