Clonal Selection based Fuzzy C-Means Algorithm for Clustering

Simone A. Ludwig
Department of Computer Science
North Dakota State University
Fargo, USA
simone.ludwig@ndsu.edu

ABSTRACT

In recent years, fuzzy based clustering approaches have shown to outperform state-of-the-art hard clustering algorithms in terms of accuracy. The difference between hard clustering and fuzzy clustering is that in hard clustering each data point of the data set belongs to exactly one cluster, and in fuzzy clustering each data point belongs to several clusters that are associated with a certain membership degree. Fuzzy c-means clustering is a well-known and effective algorithm, however, the random initialization of the centroids directs the iterative process to converge to local optimal solutions easily. In order to address this issue a clonal selection based fuzzy c-means algorithm (CSFCM) is introduced. CSFCM is compared with the basic Fuzzy C-Means (FCM) algorithm, a genetic algorithm based FCM (GAFCM) algorithm, and a particle swarm optimization based FCM (PSOFCM) algorithm.

Categories and Subject Descriptors

I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search

Keywords

Evolutionary computation, fuzzy c-means algorithm, data clustering

1. INTRODUCTION

Data mining is a relatively broad field that deals with the automatic knowledge discovery from databases, and is one of the most developed fields in the area of artificial intelligence. Given the rapid growth of data collected in various realms of human activity and their potential usefulness requires efficient tools to extract and make use of the potentially gathered knowledge [1]. One of the important data mining tasks is classification, which is an effective method that is used in many different fields. The main idea behind the classification task is to build a model (classifier) that assigns items in a collection to target classes with the goal to accurately predict the target class for each item in the data [2]. There are many techniques that can be used to do a classification process such as decision trees, Bayes networks, genetic algorithms, genetic programming and many others [3]. Another important data mining technique used when analyzing data is clustering [4]. The main goal of clustering algorithms is to divide a set of unlabeled data objects into different groups called clusters (each group has common specifications between the group members). The cluster membership measure is based on a similarity measure. To obtain high quality clusters, the similarity measure between the data objects in the same cluster is to be maximized, and the similarity measure between the data objects from different groups is to be minimized.

There are several definitions of how a cluster can be formulated depending on the objective of clustering. In general, a cluster is a group of objects that are more similar to one another than to members of other clusters [5, 6]. The term "similarity" is defined in terms of mathematical similarity with a distance norm. Distance can be measured among the data items or as a distance from a data item to some object (prototype) of the cluster. Since the objects are usually not known beforehand, they are determined by the algorithms during the clustering steps. The objects can be of the same dimension as the data objects, or can be defined as "higher-level" geometrical objects such as linear or nonlinear functions. The performance of most clustering algorithms is influenced by the geometrical shapes and densities of the individual clusters. However, it is also influenced by the spatial relations and distances of the clusters.

Many clustering algorithms have been introduced and clustering techniques can be categorized depending on whether the subsets of the resulting classification are *fuzzy* or *crisp* (hard). In *hard clustering* an object either belongs or does not belong to a cluster. In *fuzzy clustering* however, the objects belong to several clusters exhibiting different degrees of membership. Fuzzy clustering is seen as more natural than hard clustering since the objects on the class boundaries do not need to fully belong to one of the classes. The objects are assigned membership degrees between 0 and 1.

In recent years, fuzzy based clustering approaches have shown to outperform state-of-the-art hard clustering algorithms in terms of accuracy. Fuzzy c-means clustering [5] is a common and effective algorithm, however, the random initialization of the centroids directs the iterative process to converge to local optimal solutions easily. Therefore, evolutionary algorithms and swarm intelligence techniques

GECCO'14, July 12–16, 2014, Vancouver, BC, Canada.
ACM 978-1-4503-1964-5/14/07.
http://dx.doi.org/10.1145/2576768.2598270.

have been successfully applied such as genetic algorithms, ant colony optimization, and particle swarm optimization in order to tackle this problem.

This paper proposes another evolutionary algorithm technique belonging to the category of artificial immune systems. A clonal selection mechanism is combined with the fuzzy c-means algorithm. The paper is structured as follows: Section 2 presents related work starting with general categories of fuzzy clustering and ending with a list of work related to using evolutionary methods for fuzzy clustering. In Section 3, first the fuzzy c-means algorithm and the clonal selection algorithm are introduced before the proposed method is described. Section 4 lists the experimental setup and the data sets used. In Section 5, the results of the experiments are given and discussed. Section 6 concludes this paper with a summary of the findings.

2. RELATED WORK

Due to the algorithmic approach, fuzzy clustering can be categorized into three categories: hierarchical fuzzy clustering methods, graph-theoretic fuzzy clustering methods and fuzzy clustering based on objective functions [7]. Hierarchical clustering methods correspond to the determination of "similarity" trees, which is based on fuzzy equivalence relations.

Hierarchical clustering methods generate a hierarchy of partitions by means of agglomerative and divisive methods [7]. The agglomerative algorithms produce a sequence of clusters of decreasing number at each step merging two clusters from the previous level. The divisive algorithms work the other way around. Lee [8] proposed a hierarchical clustering algorithm to cluster business processes identified during business systems planning. The best number of clusters is determined by a matching approach. Another technique called fuzzy equivalent relation-based hierarchical clustering method deals with the cluster problem without a predefined number of clusters [9].

Graph-theoretic fuzzy clustering methods are based on the idea of connectivity of nodes of a graph representing the data set. In graph-theoretic fuzzy clustering, the graph representing the data structure is a fuzzy graph and different notions of connectivity lead to different types of clusters. The idea of fuzzy graphs is first mentioned in [10] whereby the fuzzy analogues of several basic graph-theoretic concepts such as bridges, cycles, paths, trees are introduced. In [11], fuzzy graphs were first used for cluster analysis.

Fuzzy clustering based on objective functions results in the most precise formulation of the clustering. The fuzzy C-Means clustering model (FCM) was first introduced in 1974 [12], and later extended and generalized in [5]. Since then, some variations of the method and model improvements are suggested.

The Gustafson-Kessel (GK) algorithm [13] is a fuzzy clustering technique that can estimate local covariance and partition data into subsets, which can be well fitted with linear submodels. However, considering a general structure for the covariance matrix can have substantial effect on the modeling approach, and therefore, the Gath-Geva algorithm [14] was proposed. The Fuzzy C-Varieties (FCV) [15] clustering algorithm is a fuzzy clustering algorithm where the prototype of each cluster is a multi-dimensional linear variety.

By replacing the Euclidean distances with other distance measures and enriching the cluster prototypes with further parameters, other shapes than just the spherical clusters can be discovered. Clusters might be ellipsoidal, linear, manifolds, quadrics or even differ in volumes [7]. Fuzzy clustering has been proven to handle ambiguous data that share properties of different clusters using membership degrees to assign data objects.

The use of fuzzy clustering especially the FCM algorithm has been shown to be effective in image segmentation [16]. However, the FCM algorithm still lacks enough robustness to noise and outliers, especially in absence of prior knowledge of noise. The time of segmenting an image depends on the image size, and hence the larger the size of the image, the longer the segmentation time [17].

Evolutionary algorithms and swarm intelligence techniques have been successfully applied such as Genetic Algorithms (GA), Ant Colony Optimization (ACO), and Particle Swarm Optimization (PSO). The key features of these evolutionary and swarm intelligence based algorithms compared to other global optimization techniques are their swarm-based collective learning ability, flexibility and robustness.

Many evolutionary computation methods have been applied for clustering. A hybrid technique based on combining the k-means algorithm, Nelder-Mead simplex search, and PSO was applied for cluster analysis in [18]. Another algorithm based on the combination of GA, k-means and logarithmic regression expectation maximization [19] was introduced. An introduced k-means algorithm that performs correct clustering without preassigning the exact number of clusters was proposed in [20]. A genetic k-means algorithm for cluster analysis was introduced in [21], and a GA based method to solve the clustering problem and experiment on synthetic and real life data sets to evaluate the performance was proposed in [22] - a basic mutation operator specific to clustering called distance-based mutation is the novelty of this approach. A GA algorithm that exchanges neighboring centers for k-means clustering has been presented in [23]. A combination of evolutionary algorithm with a ACO algorithm for the clustering problem was introduced in [23, 24].

Artificial Immune Systems (AIS) based clustering methods have also been proposed. A so called fuzzy artificial immune system clustering approach was proposed in [25]. The approach is based on artificial immune networks and fuzzy system. The authors compared their approach with the k-means algorithm and reported better results achieved by their proposed algorithm. Another algorithm is proposed in [26]. The algorithm is based on the immune mechanism, whereby the data to be clusters is represented as the antigens, and the centroids are represented as the antibodies. The clustering is therefore driven by the generation of antibodies to recognize the antigens. The solution converges towards finding the optimal antibodies for the capture of the antigens. The results showed that the proposed algorithm increases the convergence speed by avoiding local optima. The next algorithms makes use of an immunodomaince operator that is introduced to the clonal selection algorithm in [27]. This operator allows to gain prior knowledge and the sharing of information among the different antibodies. Their method was compared with the standard FCM and a GA-based FCM algorithm. The results showed that their proposed algorithm performed better than the others, in particular in avoiding the local optimum trap. Another algorithm based on artificial immune system and ant colony

optimization was proposed in [28]. The authors proposed an Immunity-based Ant Clustering Algorithm (IACA) in order to perform the clustering task automatically by finding the correct number of clusters.

Even though AIS based approaches have been explored for data clustering in the past, however, this paper proposes an approach based on the combination of standard FCM algorithm with the clonal selection principle of AIS. The FCM algorithm is a very powerful algorithm, however, it suffers from the initialization problem easily converging to suboptimal solutions. In order to overcome this clonal selection is used applying operators such as cloning, mutation, reselection and displacement.

3. PROPOSED APPROACH

This section gives an introduction of the fuzzy c-means algorithm first followed by a discussion of the clonal selection approach. Afterwards the proposed clonal selection based fuzzy c-means algorithm is described in detail.

3.1 Fuzzy c-Means Algorithm

In fuzzy clustering each data point belongs to several clusters that are associated with a certain membership degree.

The FCM algorithm is an iterative partitional clustering technique first introduced by Dunn [12], and was further extended by Bezdek [5]. FCM is a standard least squared error model that generalizes an earlier and very popular non-fuzzy c-means model that produces hard clusters of the data. An optimal c partition is produced iteratively by minimizing the *weighted within group sum of squared error* objective function:

$$J = \sum_{i=1}^{n}\sum_{j=1}^{c}(u_{ij})^m d^2(y_i, c_j) \quad (1)$$

where $Y = [y_1, y_2, ..., y_n]$ is the data set in a d-dimensional vector space; n is the number of data items; c is the number of clusters which is defined by the user where $2 \leq c \leq n$. u_{ij} is the degree of membership of y_i in the j^{th} cluster; m is a weighted exponent on each fuzzy membership; c_j is the center of cluster j; $d^2(x_i, c_j)$ is a squared distance measure between object y_i and cluster c_j.

A solution with c partitions can be obtained via an iterative process which is as follows:

1. Input c, m, threshold value ϵ, Y.
2. Initialize the fuzzy partition matrix $U = [u_{ij}]$.
3. Iteration starts by setting $t = 1$.
4. Calculate the c cluster centers with U^t:
$$c_i = \frac{\sum_{i=1}^{n}(u_{ij})^m y_i}{\sum_{i=1}^{n}(u_{ij})^m} \quad (2)$$
5. Calculate the membership U^{t+1} using:
$$u_{ij} = \frac{1}{\sum_{k=1}^{c}(\frac{d_{ij}}{d_{kj}})^{\frac{2}{(m-1)}}} \quad (3)$$
6. If one of the stopping criteria is not met, then increment t and go to Step 4. The stopping criteria are the maximum number of iterations achieved and no significant improvement compared to the previous iteration is made based on ϵ.

3.2 Clonal Selection Algorithm

De Castro and Von Zuben developed the Clonal Selection Algorithm (CSA) [29] based on the biological clonal selection theory and the shape space model of the biological immune system. The main idea is that only cells that are capable of recognizing an antigen will proliferate. CSA is very similar to a genetic algorithm, however, CSA does not have a crossover operator. The algorithm works as follows:

- Step 1. *Initialization*: Randomly initialize a population of individuals P.
- Step 2. *Evaluation*: Present each input I to the population P, and determine its affinity with each element of P.
- Step 3. *Selection and cloning*: Select n of the highest affinity elements of P, and clone these individuals proportionally to their affinity with the antigen. The higher the affinity, the higher the number of copies.
- Step 4. *Hypermutation*: Mutate all these copies with a rate proportional to their affinity with the input pattern - the higher the affinity, the smaller the mutation rate.
- Step 5. *Receptor editing*: Add these mutated individuals to the population P, and reselect m of these maturated individuals to be kept as memory cells.
- Step 6. Repeat Steps 2-5 until a certain criterion is met.

3.3 Proposed Clonal Selection based FCM Algorithm - CS-FCM

Our proposed CSFCM algorithm uses the objective function of FCM and the principles of clonal selection in order to find the optimal centroids given the membership matrix. In particular, CSFCM is designed to find the optimal membership matrix and the centroids by minimizing Equation 1.

For the notation used in the algorithm description below, the term antibody represents the centroids, antigen represents the membership matrix, and the memory cell represents the best solution or best centroids.

The CSFCM algorithm works as follows:

- **Step 1**: The antibody population is randomly generated.
- **Step 2**: The first iteration of the algorithm begins by calculating the affinity of the antibody population using the following equation:
$$f = \frac{1}{1+J} \quad (4)$$
where f is the fitness of an antibody.
- **Step 3**: The n highest affinity antibodies are selected to compose a new set of high-affinity antibodies, and the highest affinity memory cell is found. Next follows the cloning stage. The n selected antibodies are cloned based on their antigenic affinities to generate the clone set C. The number of clones for an antibody is fixed to

$n+1$. Therefore, the total number of clones generated n_c is defined as:

$$n_c = \sum_{i=1}^{n}(n+1) = n^2 + n \quad (5)$$

This function allows the optimization to get closer to the solution by increasing the average affinity.

- **Step 4**: The next step is mutation. Each antibody in the clone set C gets the opportunity to produce mutated offspring abiding by the law that the higher the affinity, the smaller the mutation rate. In order to achieve this, the affinity of the antibodies are normalized (f_{norm}) to be in the range of [0,1]. The mutation rate is adaptively determined by:

$$p_m = exp(-2 \times f_{norm}(antibody)) \quad (6)$$

where p_m is the mutation rate (in the range of [0,1]), and 2 is the decay coefficient. The mutation process is a very important step in the algorithm. It generates a random real value using a uniform distribution (u) in the range between the minimum and the maximum. The function is defined as:

$$\Delta(I, u) = u(1 - r^{(1-i/I)^{\lambda}}) \quad (7)$$

where r is a random value in the range of [0,1], i is the current iteration, I is the maximum number of iterations defined, and λ is the nonconforming degree factor. This step is crucial for the algorithm since the mutation step helps to avoid local optima. One antibody is kept in order to keep the search stable. Then, the affinity of the matured clones is calculated. Afterwards, the antibodies with the highest affinity are replacing the ones with the lowest affinity, and the one with the highest affinity is set to be the memory cell. During the iteration, this memory cell is only updated if there was an improvement compared to the previous iteration. Then, the antibodies with the lowest affinity are replaced by newly randomly generated antibodies.

- **Step 5**: The stopping criterion is the maximum number of iterations reached. If the stopping criterion is not fulfilled, then the algorithm proceeds with Step 2.

- **Step 6**: Calculate the validity indices (described in Section 4.2).

4. EXPERIMENTAL SETUP

This section describes the parameter setup and the data sets used as well as describes the validity measures proposed in literature that are being used for the experiments.

4.1 Parameter Setup and Data sets

The parameters of the CSFCM algorithm are set to:

- Population size (antibodies) = 20
- Maximum number of iterations = 100
- Number of selected antibodies = 8
- Number of displaced antibodies = 5
- Fuzzy weighting exponent = 2

The data sets used for the experiments are given in Table 1. The number of records, dimensions, and number of clusters are listed. 30 independent runs were performed for each data set.

Table 1: Data sets

Data set	Records	Dimensions	Clusters
Iris	150	4	3
Wine	178	13	3
Vowel	871	3	6
Glass	214	10	6
Ecoli	336	7	8
Liver disorder	345	7	2
Vowel2	528	10	11

4.2 Validity Indices

To measure the quality of the resulting clusters, several cluster validity measures have been proposed in literature. The cluster validity is a measure of the relative performance of a partitioned structure of the data set. All clustering algorithms generate a partition matrix and other useful information regarding the cluster structure by identifying centroids. Partition and centroids jointly determine the "goodness" of a cluster structure. The validity indices used in this study are explained below.

4.2.1 Partition Coefficient (PC) Index

The Partition Coefficient (PC) is defined as [5]:

$$PC = \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{c} u_{ij}^2 \quad (8)$$

PC obtains its maximum when the cluster structure is optimal.

4.2.2 Partition Entropy (PE) Index

The Partition Entropy (PE) is defined as [15]:

$$PE = -\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{c} u_{ij} log_b(u_{ij}) \quad (9)$$

where b is the logarithmic base. PE obtains its minimum when the cluster structure is optimal.

4.2.3 Davies-Bouldin (DB)

The Davies-Bouldin (DB) fuzzy validity index is defined as the ratio of the sum of within-cluster scatter to between-cluster separation [30]:

$$DB = \frac{1}{c}\sum_{i=1}^{c} R_i \quad (10)$$

where c is the number of clusters, $R_i = \max_{i \neq j}\{S_i + S_j)/l_{ij}\}$. The scatter with the j^{th} center is $S_j = (1/|C_j|)\sum_{x \in C_i} ||y - z_j||^2$. The distance between the i^{th} cluster center and the j^{th} cluster center is $l_{ij} = ||z_i - z_j||^2$. DB achieves the minimum when the cluster structure is optimal.

4.2.4 Partition Coefficient And Exponential Separation (PCAES) Index

The Partition Coefficient And Exponential Separation (PCAES) index [31] is defined as:

$$PCAES = \sum_{i=1}^{n}\sum_{j=1}^{c}\frac{(u_{ij})^2}{u_M} - \sum_{k=1}^{c} exp(-\min_{k \neq i}||z_i - z_k||^2/\beta_T) \quad (11)$$

where $u_M = \min_{1 \leq j \leq c}\{\sum_{i=1}^{n} u_{ij}^2\}$ and $\beta_T = (\sum_{j=1}^{c} ||z_j - \bar{z}||^2)/c$. $\bar{z} = \sum_{i=1}^{n}(y_i/n)$. PCAES reaches its maximum when the cluster structure is optimal.

4.2.5 Pakhira-Bandyopadhyay-Maulik Fuzzy (PBMF) index

The Pakhira-Bandyopadhyay-Maulik Fuzzy (PBMF) index for fuzzy clustering is defined as [32]:

$$PBMF = (\frac{1}{c} \times \frac{E_1}{J_m} \times D_c)^2 \quad (12)$$

where

$$E_1 = \sum_{i=1}^{n} ||y_i - z|| \quad (13)$$

E_1 is a constant determined by the data set, with z being the centroid of the data set.

$$D_c = \max_{i,j=1}^{c} ||z_i - z_j|| \quad (14)$$

PBMF obtains its maximum when the cluster structure is optimal.

4.2.6 Xie-Beni (XB) Index

Xie and Beni proposed a validity function in 1991 [33] and later it was modified by Bezdek in 1995 [34].

$$XB = \frac{J_m}{n \times \min_{i \neq j} ||z_i - z_j||^2} \quad (15)$$

XB reaches its minimum when the cluster structure is optimal.

5. RESULTS

The results of applying the different validity indices are shown in Tables 2-7. The comparison values for FCM, GAFCM and PSOFCM are taken from [35]. The comparison algorithms used the following parameters. FCM was run for 100 iterations. GAFCM used a population size of 50 chromosomes, a crossover rate of 0.85, and a mutation rate of 0.008, and was run for 100 iterations. PSOFCM used a swarm size of 50 particles, the maximum and minimum inertia weight were set to 0.9 and 0.4, respectively, the constants c_1 and c_2 were both set to 2, with the number of iterations set to 120.

Table 2 lists the comparison of FCM, GAFCM, PSOFCM and CSFCM in terms of the PC validity index applied to the given data sets. It can be seen that the CSFACM algorithm achieves 3 out of 7 times the best PC validity index tied with the FCM algorithm. GAFCM as well as PSOFCM achieve the highest PC index on one data set each.

Looking at Table 3, the PE validity index for all algorithms measured on the data sets is given. It shows that CSFCM achieves the lowest values and therefore best values for 4 data sets, again tied with the FCM algorithm.

From Table 4 it can be seen that the CSFACM algorithm achieves the best DB validity value 5 out of 7 times, whereas FCM obtains the best index in 4 cases, and GAFCM only in 1.

The PCAES validity index values are given in Table 5. Our CSFCM algorithm outperforms the other techniques scoring the highest PCAES validity values on 5 data sets. The PSOFCM algorithm scores best on 2 data sets.

CSFCM obtains the highest PBMF validity index on 6 data sets as seen in Table 6. FCM scores the best value on 3 data sets, and PSOFCM obtains the highest score on the Liver disorder data set.

Table 7 contains the XB validity values. Again, CSFCM scores best obtaining the lowest and therefore best score on 5 data sets, whereas FCM scores best 3 times. GAFCM and PSOFCM achieve the best value on 1 data set each.

Overall, our CSFCM algorithm obtains the better validity values for 4 of the 6 validity indices. For the other 2, CSFCM is tied with FCM.

Figure 1 summarizes the results graphically. It can be seen by the number of wins, draws, and losses that CSFCM outperforms the other algorithms followed by FCM. A win is defined as having the best validity index (highest or lowest) compared to the other algorithms, a loss is defined as not having the best validity index, and a draw is defined if two or more algorithms achieved exactly the same best validity index. The number of wins, draws and losses are summed over all data sets.

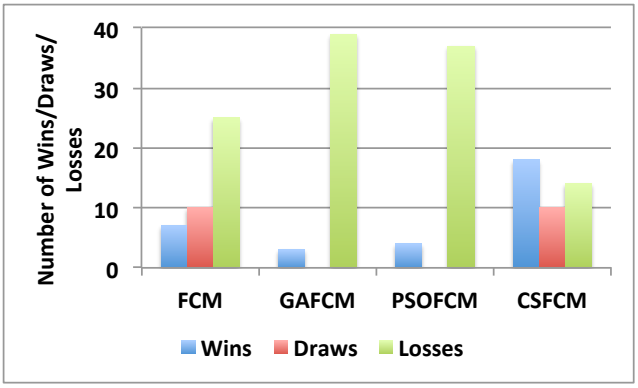

Figure 1: Number of wins, draws, and losses of all algorithms

However, it has to be mentioned that our CSFCM algorithm used the highest number of Function Evaluations (FE) with 9,700 FE (20 antibodies plus 72 clones plus 5 displacement antibodies times 100 iterations). The FCM achieving very good results only needed 100 FE, and GAFCM used 5,000 FE (50 chromosomes times 100 iterations), and PSOFCM ran with 6,000 FE (50 particles times 120 iterations).

6. CONCLUSION

Fuzzy based clustering approaches have shown to outperform state-of-the-art hard clustering algorithms. The difference between hard clustering and fuzzy clustering is that in hard clustering each data point of the data set belongs to exactly one cluster, and in fuzzy clustering each data point belongs to several clusters that are associated with a certain membership degree. Fuzzy c-means clustering is a well-known and effective algorithm, however, the random initialization of the centroids directs the iterative process to converge to local optimal solutions easily. This paper proposed a clonal selection based fuzzy clustering algorithm (CSFCM) in order to overcome this issue. The algorithm makes use of clonal selection theory by applying methods such as cloning, mutation, reselection and displacement as well as using the objective function of the FCM algorithm.

Seven data sets were chosen and the CSFCM algorithm was compared to the basic fuzzy c-means algorithm (FCM),

Table 2: Comparison of FCM, GAFCM, PSOFCM, CSFCM in terms of PC validity index

Data set	FCM	GAFCM	PSOFCM	CSFCM
Iris	**0.7832±3.7633e-16**	0.75823±0.01351	0.77417±0.0099931	0.7618±1.91761e-5
Wine	**0.79094±1.1.4057e-12**	0.79004±0.0011282	0.78845± 0.0024337	0.79004±1.6337-7
Vowel	0.54976±9.3088e-8	0.49496±0.018895	0.52674± 0.012648	**0.55004±4.2974e-8**
Glass	0.50173±0.006688	**0.51232±0.059044**	0.49461±0.095027	0.50827±7.1837e-9
Ecoli	0.30873±0.001027	0.27437±0.011477	**0.37804±0.031598**	**0.37804±6.1944e-6**
Liver disorder	0.82993±3.7026e-16	0.82488±0.01628	0.81221 ±0.080893	**0.83103±2.4917e-8**
Vowel2	**0.22873±0.0013672**	0.12554±0.0035406	0.18685± 0.0055172	0.2182±5.2344e-6

Table 3: Comparison of FCM, GAFCM, PSOFCM, CSFCM in terms of PE validity index

Data set	FCM	GAFCM	PSOFCM	CSFCM
Iris	**0.39593±3.341e-16**	0.43395±0.021827	0.4133±0.018546	0.40857±1.349e-9
Wine	**0.38041±1.5985e-12**	0.38204±0.0015792	0.3855±0.0049499	**0.38041±4.8934e-8**
Vowel	**0.92244±9.6191e-7**	1.0216±0.034679	0.97007±0.026595	0.92293±7.6243e-7
Glass	0.96943±0.02053	1.4308±0.12146	1.0842±0.17089	**0.9479±2.9477e-8**
Ecoli	1.5333±0.011206	1.9196±0.034903	1.6449±0.082692	**1.5296±1.2673e-4**
Liver disorder	**0.28813±2.7948e-16**	0.29613±0.023051	0.30998 ±0.11027	0.29017±1.3728e-7
Vowel2	1.8882±0.0032484	2.228±0.014348	1.5143±0.027739	**1.4378±3.8373e-5**

a genetic algorithm based FCM algorithm (GAFCM), as well as a particle swarm optimization based FCM algorithm (PSOFCM). Six validity indices were used in order to compare the algorithms with each other. The results showed that CSFCM outperformed the other algorithms for four of the validity indices, and scored equally well compared to FCM on the other two validity indices in terms of best validity value achieved.

Future work will address the second shortcoming of FCM with the number of clusters needing to be predefined. Furthermore, in order for the algorithm to be applicable on big data sets, a parallelization of the algorithm is paramount.

7. REFERENCES

[1] A. Ghosh and L. C. Jain, "Evolutionary Computation in Data Mining Series: Studies in Fuzziness and Soft Computing", vol. 163, Sprinter, 2005.

[2] P. Tan, M. Steinbach, and V. Kumar, "Introduction to Data Mining", Addison-Wesley, May 2005.

[3] H. Jabeen and A. R. Baig. "Review of Classification Using Genetic Programming", International Journal of Engineering Science and Technology, vol. 2, no. 2, pp. 94-103, 2010.

[4] J. Han, "Data Mining: Concepts and Techniques", Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2005.

[5] J. C. Bezdek, "Pattern Recognition with Fuzzy Objective Function Algorithms", Kluwer Academic Publishers Norwell, MA, USA, 1981.

[6] A. K. Jain and R. C. Dubes, "Algorithms for Clustering Data, Prentice-Hall, Inc., Upper Saddle River, NJ, 1988.

[7] M. S. Yang, "A survey of fuzzy clustering", Math. Comput. Modelling, 18, pp. 1-16, 1993.

[8] H. S. Lee, "Automatic clustering of business process in business systems planning", European Journal of Operational Research, 114, pp. 354-362, 1999.

[9] G. J. Klir, B. Yuan, "Fuzzy Sets and Fuzzy Logic Theory and Application", Prentice Hall PTR, Upper Saddle River, NJ, 1995.

[10] A. Rosenfeld, Fuzzy graphs, in: L. A. Zadeh, K. S. Fu, M. Shimura (Eds.), "Fuzzy Sets and their Applications to Cognitive and Decision Processes", Academic Press, New York, 1975.

[11] D. W. Matula, "Cluster analysis via graph theoretic techniques", Proceedings of the Louisiana Conference on Combinatorics, Graph Theory and Computing, Winnipeg, 1970.

[12] J. C. Dunn, "A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters", Journal of Cybernetics 3: 32-57, 1973.

[13] V. P. Guerrero-Bote, C. Lopez-Pujalte, F. de Moya-Anegon, V. Herrero-Solana, "Comparison of neural models for document clustering", Int. Journal of Approximate Reasoning, vol. 34, pp. 287-305, 2003.

[14] I. Gath and A. B. Geva, "Unsupervised optimal fuzzy clustering", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 11(7), pp. 773-781, 1989.

[15] J. C. Bezdek, C. Coray, R. Gunderson and J. Watson, "Detection and Characterization of Cluster Substructure - Linear Structure, Fuzzy c-Varieties and Convex Combinations Thereof", SIAM J. Appl. Math., vol. 40, no. 2, pp. 358-372, 1981.

[16] Y. Yang and S. Huang, "Image Segmentation by Fuzzy C-Means Clustering Algorithm with a Novel Penalty Term", in Computing and Informatics, vol. 26, pp. 17-31, 2007.

[17] W. Cai, S. Chen and D. Zhang, "Fast and robust fuzzy c-means clustering algorithms incorporating local information for image segmentation", Pattern Recognition, vol. 40, no. 3, pp. 825-838, 2007.

[18] Y. T. Kao, E. Zahara, I. W. Kao, "A hybridized approach to data clustering", Expert Systems with Applications 34 (3), 1754-1762, 2008.

Table 4: Comparison of FCM, GAFCM, PSOFCM, CSFCM in terms of DB validity index

Data set	FCM	GAFCM	PSOFCM	CSFCM
Iris	**0.91204±1.0549e-15**	0.92938±0.0086253	0.92095±0.010754	0.91317±1.8342e-8
Wine	**116.44±1.4355e-14**	116.68±0.77969	116.66±0.86855	**116.44±6.5433e-12**
Vowel	**187.6±1.7698e-14**	193.78±0.0731	192.13±0.6523	**187.6±2.17236e-11**
Glass	2.179±0.029096	2.425±0.5164	2.673±0.645	**2.1714±2.7583e-13**
Ecoli	3.9223±0.46362	**3.851±0.623**	4.0258±0.704	3.8552±3.0548e-6
Liver disorder	**45.843±0**	45.967±0.57959	46.641±3.3651	**45.843±0**
Vowel2	2.3076±0.035978	2.632±0.1304	2.512±0.2164	**2.189±4.2840e-3**

Table 5: Comparison of FCM, GAFCM, PSOFCM, CSFCM in terms of PCAES validity index

Data set	FCM	GAFCM	PSOFCM	CSFCM
Iris	1.972±0.52633	1.8809±0.46614	**2.33±0.89747**	2.1348±4.2327e-4
Wine	2.2064±0.77077	2.0673±0.80591	2.1656±0.86423	**2.1361±4.5722e-5**
Vowel	2.0828±0.95751	2.7142±0.2415	2.202±0.724	**2.7902±3.4975e-3**
Glass	8.0692±0.1788	9.109±0.636	9.047±0.638	**9.1543±3.9547e-4**
Ecoli	2.822±0.7289	2.3986±0.7279	2.911±0.222	**2.9159±7.5678e-3**
Liver disorder	3.136±0.104	3.9429±0.4519	2.8944±0.2186	**4.2523±2.2852e-3**
Vowel2	3.0307±0.1848	3.8321±0.9351	**8.0544±0.2493**	4.5275±4.1387e-4

Table 6: Comparison of FCM, GAFCM, PSOFCM, CSFCM in terms of PBMF validity index

Data set	FCM	GAFCM	PSOFCM	CSFCM
Iris	**1.1343±4.7263e-16**	1.0228±0.065783	1.0807±0.06328	**1.1343±3.4721e-12**
Wine	**1.0407±1.0589e-12**	1.0353±0.004563	1.0248±0.01489	**1.0407±5.2125e-11**
Vowel	**0.15345±4.2779e-7**	0.14099±0.0066558	0.15033±0.005635	**0.15345±7.2195e-9**
Glass	0.56154±0.0037025	0.28959±0.04661	0.42935±0.054079	**0.5874±2.6235e-8**
Ecoli	0.24857±0.0029411	0.15323±0.017463	0.24285±0.048303	**0.2278±2.6487e-4**
Liver disorder	0.28445±2.5327e-16	0.27592±0.021256	**0.29362±0.066964**	0.27364±6.1446e-8
Vowel2	0.15191±0.0019245	0.12853±0.0056154	0.12436±0.026612	**0.16173±1.7485**

Table 7: Comparison of FCM, GAFCM, PSOFCM, CSFCM in terms of XB validity index

Data set	FCM	GAFCM	PSOFCM	CSFCM
Iris	**0.13711±5.9074e-16**	0.20954±0.04634	0.142±0.01772	**0.13711±5.3491e-9**
Wine	0.12566±3.6067e-12	**0.12522±0.001930**	0.12846±0.00484	0.12582±5.3189e-10
Vowel	**0.18933±4.492e-8**	0.48874±0.20922	0.25009±0.074111	**0.18933±3.7648e-10**
Glass	0.94653±0.047904	1.1628±0.57795	**0.31815±0.21906**	0.91448±2.6478e-8
Ecoli	0.91204±0.08645	1.6848±0.66646	0.90928±1.0236	**0.8013±4.1698e-5**
Liver disorder	**0.12617±1.2928e-16**	0.13726±0.034688	0.14235±0.09482	**0.12617±2.1773e-10**
Vowel2	0.60571±0.2154	1.6359±0.50056	1.2361±0.2064	**0.5073±6.1847e-4**

[19] D. N. Cao, J. C. Krzysztof, "GAKREM: a novel hybrid clustering algorithm", Information Sciences 178, 4205-4227, 2008.

[20] K. R. Zalik, "An efficient k-means clustering algorithm", Pattern Recognition Letters 29, 1385-1391, 2008.

[21] K. Krishna, Murty, "Genetic k-means algorithm", IEEE Transactions of System Man Cybernetics Part B-Cybernetics 29, 433-439, 1999.

[22] U. Mualik, S. Bandyopadhyay, "Genetic algorithm-based clustering technique", Pattern Recognition 33, 1455-1465, 2000.

[23] M. Laszlo, S. Mukherjee, "A genetic algorithm that exchanges neighboring centers for k-means clustering", Pattern Recognition Letters 28 (16), 2359-2366, 2007.

[24] P. S. Shelokar, V. K. Jayaraman, B. D. Kulkarni, "An ant colony approach for clustering", Analytica Chimica Acta 509 (2), 187-195, 2004.

[25] Z. Liu, X. Jin, R. Bie, X. Gao, "FAISC: A Fuzzy Artificial Immune System Clustering Algorithm", Proceedings of the Third International Conference on Natural Computation (ICNC), 2007.

[26] T. Liu, Y. Zhou, Z. Hu, Z. Wang, "A New Clustering Algorithm Based on Artificial Immune System", Proceedings of the Fifth International Conference on Fuzzy Systems and Knowledge Discovery, 2008.

[27] R. Liu, Z. Shen, L. Jiao, W. Zhang, "Immunodomaince based clonal selection clustering algorithm", Proceedings of the 2010 IEEE Congress on Evolutionary Computation (CEC), 2010.

[28] C. Chui-Yu, C.-H. Lin, "Cluster Analysis Based on Artificial Immune System and Ant Algorithm", Proceedings of the Third International Conference on Natural Computation, 2007.

[29] L. N. De Castro and F. J. Von Zuben, "Learning and optimization using the clonal selection principle", IEEE Transactions on Evolutionary Computation, vol. 6, no. 3, pp. 239-251, 2002.

[30] D. L. Davies and D. W. Bouldin, "A cluster separation measure", IEEE Transactions on Pattern Analysis and Machine Intelligence, (2), 224-227, 1979.

[31] K. L. Wu, and M. S. Yang, "A cluster validity index for fuzzy clustering", Pattern Recognition Letters, 26(9), 1275-1291, 2005.

[32] M. K. Pakhira, S. Bandyopadhyay, U. Maulik, "Validity index for crisp and fuzzy clusters", Pattern Recognition, 37, 487-501, 2004.

[33] X. L. Xie, and G. Beni, "A validity measure for fuzzy clustering", IEEE Transactions on Pattern Analysis and Machine Intelligence, 13(8), 841-847, 1991.

[34] N. R. Pal, and J. C. Bezdek, "On cluster validity for the fuzzy c-means model", IEEE Transactions on Fuzzy Systems, 3(3), 370-379, 1995.

[35] C. Li, J. Zhou, P. Kou, J. Xiao, "A novel chaotic particle swarm optimization based fuzzy clustering algorithm", Journal of Neurocomputing, Elsevier, vol. 83, no. 1, 2012.