Journal of Theoretical and Applied Information Technology
30th November 2014. Vol. 69 No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

DATA CLUSTERING BASED ON HYBRID OF FUZZY AND SWARM INTELLIGENCE ALGORITHM USING EUCLIDEAN AND NON-EUCLIDEAN DISTANCE METRICS: A COMPARATIVE STUDY

[1]O.A. MOHAMED JAFAR, [2]R.SIVAKUMAR

[1]Research Scholar, Department of Computer Science, A.V.V.M. Sri Pushpam College (Autonomous), Poondi, Thanjavur, Tamil Nadu, India

[2]Associate Professor, Department of Computer Science, A.V.V.M. Sri Pushpam College (Autonomous), Poondi, Thanjavur, Tamil Nadu, India

E-mail: [1]jafarmdoa@yahoo.in, [2]rskumar.avvmspc@gmail.com

ABSTRACT

Data mining is a collection of techniques used to extract useful information from large data bases. Data clustering is a popular data mining technique. It is the task of grouping a set of objects into classes such that similar objects are placed in the same cluster while dissimilar objects are in separate clusters. Fuzzy c-means (FCM) is one of the most popular clustering algorithms. However, it has some limitations such as sensitivity to initialization and getting struck at local optimal values. Swarm intelligence algorithms are global optimization techniques and are recently successfully applied to solve many real-world optimization problems. Constriction Factor Particle Swarm Optimization (cfPSO) algorithm is a population based global optimization technique which is used to solve data clustering problems. Euclidean distance is a well known and commonly used metric in most of the literature. Some drawbacks of this distance metric include blind to correlated variables, not robust in noisy environment, affected by outlier data points and handle data sets with only equal size, density and spherical shapes. But real-world data sets may exhibit different shapes. In this paper, a Fuzzy based Constriction Factor PSO (FUZZY-cfPSO-FCM) algorithm is proposed using Non-Euclidean distance metrics such as Kernel, Mahalanobis and New distance on several benchmark UCI machine learning repository data sets. The proposed hybrid algorithm makes use of the advantages of FCM and cfPSO algorithms. The clustering results are also evaluated through fitness value, accuracy rate and failure rate. Experimental results show that proposed hybrid algorithm achieves better result on various data sets.

Keywords: *Data Clustering, Fuzzy c-means (FCM), Swarm Intelligence (SI), Constriction Factor Particle Swarm Optimization (cfPSO), Euclidean Distance Metric, Non-Euclidean Distance Metrics*

1. INTRODUCTION

Data mining is the process of analyzing the data sets and then extracting the useful information. It is one of the steps in knowledge discovery process. It uses techniques such as pattern recognition, neural networks, artificial intelligence, and statistics. Important data mining tasks are class description, association, classification, clustering, prediction, trend analysis, and time-series analysis. Data clustering is a method of partition the given data set into clusters so that the data in each cluster share similar patterns. It has been used in several fields such as data mining, pattern recognition, statistics, machine learning, bioinformatics, information retrieval, and image segmentation [1].

In hard data clustering algorithms, each data point belongs to only one cluster. Hard clustering algorithms require the prior information about the number of clusters. These methods are not appropriate for real world data sets in which there are no definite boundaries between the clusters. K-means [2] and K-medoids [3] are popular classical hard clustering algorithms. Fuzzy c-means (FCM) algorithm proposed by Dunn [4] and then generalized by Bezdek [5]. In fuzzy clustering, the data points can belong to more than one cluster with different degrees of membership values. The

Journal of Theoretical and Applied Information Technology
30th November 2014. Vol. 69 No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

traditional clustering algorithms have some drawbacks such as sensitive to initial values and easily trapped in local optimal values.

In order to overcome the problems of traditional clustering algorithms, swarm intelligence based techniques have been developed in the literature. Particle Swarm Optimization (PSO) is a powerful swarm intelligence technique introduced by Kennedy and Eberhart in 1995 [6]. Eberhart and Shi [7] made a comparison of particle swarm optimization using an inertia weight and a constriction factor. Constriction Factor Particle Swarm Optimization (cfPSO), another variant of PSO, proposed by Clerc and Kennedy in 2002 [8]. The performance of the PSO algorithms depends on various parameters.

Recently, hybrid algorithms have been used by the researchers to improve the performance of the individual algorithms. Euclidean distance is more commonly used metric in many clustering algorithms. It is effective in clustering spherical shapes. It has some drawbacks such as cannot detect unequal size, sensitivity to noise and outlier and inability to handle data set with elliptical shapes. But in real-life applications there may be different sizes and cluster shape. In this paper, FUZZY-cfPSO-FCM is proposed using Non-Euclidean distance metrics such as Kernel, Mahalanobis and New distance. FCM, cfPSO and Hybrid algorithm are experimented on five real-world data sets, wine, vowel, liver disorders, glass and blood transfusion. The experimental results show that the proposed method works better than others.

2. PRELIMINARY

In this section, clustering methods, fuzzy c-means algorithm and swarm intelligence algorithms are briefly described.

2.1 Clustering Methods

Clustering methods can be broadly classified as partitioning, hierarchical, probabilistic and model-based methods [1][9].

Partitional methods construct partitions of a data set of 'n' objects into a set of 'k' clusters. Each object belongs to only one cluster and each cluster has at least one object.

Hierarchical methods form the cluster by partitioning the instances as agglomerative or divisive. Agglomerative methods begin with each instance forming a cluster of its own. They successively merge the instances that are close to one another, until all the groups are merged into one or until a termination condition holds. Divisive methods start with all the instances in one cluster and divide the cluster into small clusters. This procedure continues until the desired cluster structure is obtained.

Probabilistic clustering methods are an attempt to optimize the fit between the data and the model using probabilistic approach. Each cluster can be represented by Poisson, Gaussian or Mixture of these distributions.

Density-based methods are based on density such as density connected points. The density is defined as the number of objects in a particular neighborhood of the data objects.

Model-based methods attempt to optimize the best fit between the data and the given mathematical model. Two major approaches of these models are neural network approach or statistical approach.

2.2 Fuzzy c-means Algorithm

Fuzzy c-means (FCM) Algorithm [4] [5] permits one piece of data belong to two or more clusters. Given a data set $X = \{x_1, x_2 ... x_n\}$ in R^n dimensional space, the FCM algorithm partitions data set into c fuzzy clustering $(2 \le c \le n)$ with $z = \{z_1, z_2, ... z_c\}$ cluster centroids by minimizing the objective function value. u_{ij} is the fuzzy membership values or degree of association of the i-th data vector and j-th cluster and $m \in [1, \infty)$ is the fuzziness index. The characteristics of U are as follows:

$$u_{ij} \in [0,1], \forall i = 1,2,\ldots,n; \forall j = 1,2,\ldots,c \quad (1)$$

$$\sum_{j=1}^{c} u_{ij} = 1, \forall i = 1, 2, \ldots, n \quad (2)$$

$$0 < \sum_{i=1}^{n} u_{ij} < n, \forall j = 1,2,\ldots,c; 1 < n < \infty \quad (3)$$

Algorithm Steps:

Input: Data set
Output: Cluster centers, Fitness value

Step 1: Select the number of clusters c; choose the fuzziness index m (m>1); initialize fuzzy partition membership values $U^{(0)}$; iteration error ε=0.00001; Fix the maximum number of iterations *max_it*.
Step 2: Set the iteration counter t = 0

Journal of Theoretical and Applied Information Technology
30th November 2014. Vol. 69 No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Step 3: Calculate the cluster centers using

$$z_j = \frac{\sum_{i=1}^{n} u_{ij}^m x_i}{\sum_{i=1}^{n} u_{ij}^m}, j = 1,2,...,c \quad (4)$$

Step 4: Calculate d_{ij}, the distance from data vector x_i and cluster center z_j

Step 5: Calculate the fitness or objective function value J_m using $J_m = \sum_{j=1}^{c}\sum_{i=1}^{n} u_{ij}^m d_{ij}$ (5)

Step 6: Update the fuzzy partition membership values $U^{(t+1)}$ using

$$u_{ij}^{(t+1)} = \frac{1}{\sum_{k=1}^{c}\left(\frac{d_{ij}^2}{d_{ik}^2}\right)^{\frac{1}{m-1}}}, 1 \le i \le n; 1 \le j \le c \quad (6)$$

Step 7: If $\| U^{(t+1)} - U^{(t)} \| < \varepsilon$ or t = *max_it* then stop; otherwise set t = t+1 and go to step 3

2.3 Swarm Intelligence Algorithms

Swarm Intelligence Algorithms [10] are popular global nature-inspired techniques emerging from social animals, insects, birds, fish or mammals. They have been successfully applied to solve optimization problems in the areas like data mining, data clustering, network scheduling, communication networks and others. Ant Colony Optimization (ACO), Particle Swarm Optimization (PSO), Firefly Algorithm, Bat Algorithms, Honey Bee Mating, Bird Flocking Algorithm and Artificial Immune Systems are some of the swarm intelligence based systems.

2.3.1 Particle swarm optimization

Particle Swarm Optimization (PSO) is an efficient global optimization technique and introduced by Kennedy and Eberhart [6]. In recent years, PSO algorithm has been successfully applied to solve various optimization problems. It is inspired by the social behavior of animals such as a flock of birds, a swarm of bees or a school of fish. A swarm is a collection of particles. Each particle in the population is assumed to fly over the search space to find the desired areas. In a d-dimensional search space, the position of the i-th particle is given by $X_i = (x_{i1}, x_{i2}, ..., x_{id})$ and velocity of the i-th particle is defined as $V_i = (v_{i1}, v_{i2}, ..., v_{id})$. Each particle's best experience is known as the local best and minimum among entire swarm is called the global best. The local best and global best of particles are represented by $p_i = (p_{i1}, p_{i2}, .., p_{id})$ and $p_g = (p_{g1}, p_{g2}, ..., p_{gd})$ respectively. The velocities and positions are updated in each iteration step. The velocity of each particle is updated using p_i and p_g.

A single particle denotes the N cluster center in a PSO-based clustering algorithm. Each particle X is formed as follows:

$X_i = (m_{i1}, m_{i2}, ..., m_{iN_c})$

where N_c represents the number of clusters to be formed and m_{ij} denotes the j-th cluster center of the i-th particle in the cluster C_{ij}

3. REVIEW OF LITERATURE

Data clustering is a NP-hard problem. In literature, there are many methods for solving clustering problem. FCM algorithm [4] [5] is one of the most widely used partitioning techniques. It has been successfully used in many applications [11] [12] [13]. Kuo-Lung Wu and Miin-Shen Yang [14] proposed a new metric in c-means clustering algorithm. They have created two clustering methods called the alternative hard c-means (AHCM) and alternative fuzzy c-means (AFCM) clustering algorithms. The new metric is more robust than Euclidean norm. Zhang and Chen [15] proposed a kernel-based fuzzy and possibilistic c-means clustering. The proposed algorithms have advantages over the FCM and PCM algorithms. Bighnaraj Naik and Sarita Mahapatra [16] proposed a swarm intelligence based nature-inspired center-based clustering method using PSO method.

Omran et al. [17] proposed a new image classification algorithm based on particle swarm optimization. The algorithm is used to find the centroids of a user specified number of clusters, where each cluster groups together similar pixels. Van der Merwe and Engelbrecht [18] tested two algorithms, namely a standard gbest PSO and hybrid approach. In their paper, the individuals of the swarm are seeded by the result of K-means algorithm. Esmin et al. [19] proposed two new data clustering approaches using particle swarm optimization algorithm. Mohamed Jafar and Sivakumar [20] presented a study of particle swarm optimization algorithm to data clustering using different distance measures such as Euclidean, Manhattan and Chebyshev.

Journal of Theoretical and Applied Information Technology
30th November 2014. Vol. 69 No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Kao and Lee [21] presented a dynamic data clustering algorithm by combining K-means and particle swarm optimization. Fun Ye and Ching-Yi Chen [22] developed a hybrid PSO and K-means algorithm, called Alternative KPSO clustering (AKPSO) for automatically detect the cluster centers. They have presented an evolutionary particle swarm optimization learning-based method to optimally cluster N data points into K cluster. Cui and Potok [23] presented a hybrid particle swarm optimization (PSO) + K-means algorithm for document clustering. The problems involved in K-means algorithm is avoided in their hybrid algorithm. Niknam et al. [24] proposed an efficient hybrid approach on PSO, ACO and K-means called PSO-ACO-K approach for clustering analysis. Mehdi Neshat et al. [25] presented a cooperative algorithm based on PSO and k-means for data clustering problem. The algorithm is utilized both global search ability of PSO and local search ability of k-means. Li Yi Ran et al. [26] proposed the K-means based on Chaos Particle Swarm Optimization (CPSOKM). The algorithm is used to solve the problem involved in K-means method.

Pang et al. [27] presented a fuzzy discrete particle swarm optimization (FPSO) for solving travelling salesman problem. Runkler and Katz [28] introduced the two new methods for minimizing the two reformulated versions of the FCM objective function by particle swarm optimization (PSO). Mehdizadeh et al. [29] presented an efficient hybrid method, fuzzy particle swarm optimization (FPSO) and fuzzy c-means (FCM) to solve the fuzzy clustering problem. The hybrid algorithm is compared with FCM algorithm using different data sets. Izakian and Abraham [30] proposed a hybrid fuzzy clustering method based on Fuzzy c-means (FCM) and Fuzzy Particle Swarm Optimization (FPSO). In their algorithm, the merits of FCM and PSO have been considered. Yong Zhang et al. [31] proposed Image segmentation using PSO and PCM with Mahalanobis distance. In their algorithm, the Euclidean distance is replaced by Mahalanobis distance in the possibilistic c-means clustering algorithm with optimizing the initial clustering centers using particle swarm optimization. Chaoshun Li et al. [32] proposed a novel chaotic particle swarm fuzzy clustering (CPSFC) algorithm based on chaotic particle swarm (CPSO) and gradient method. In their algorithm, they have introduced adaptive inertia weight factor (AIWF) and iterative chaotic map with infinite collapses (ICMIC).

4. METHODOLOGY

Hybrid algorithms are the integration of two or more optimization techniques. Nowadays hybrid algorithms are popular due to the ability in handing many real-life problems that involve uncertainty and difficulty. Hybridization is used to overcome of the problems involved in individual algorithms. Distance metrics are played an important role in data clustering. They are used to find the distance between data point and cluster center. Most clustering algorithms are based on Euclidean distance metric. Euclidean distance metric is computed as follows:

$$d(x,z) = \sqrt{\sum_{i=1}^{d}(x_i - z_i)^2} = \| x_i - z_i \| \quad (7)$$

Some of the limitations of this distance metric are such as inability to handle data sets with noise and outlier data points, clusters with elliptical shapes and sensitive to scales of the variables involved.

4.1 Hybrid Algorithms Using Non-Euclidean Distance Metrics

In the literature, there are many hybrid algorithms for solving optimization problems [27]-[32]. In this paper, we have made a study of integrating FUZZY-cfPSO with FCM algorithm called (FUZZY-cfPSO-FCM) using Non-Euclidean distance metrics such as Kernel, Mahalanobis and New distance. The performance of the algorithms is evaluated through fitness value, accuracy rate and error rate. The fitness value of cfPSO algorithm is calculated by the equation

$$\sum_{j=1}^{N_c}\sum_{\forall z_p \in C_{ij}} d\left(z_p, m_j\right) \quad (8)$$

where

$$d\left(z_p, m_j\right) = \min_{\forall c = 1, \ldots, N_c}\{d(z_p, m_{ic})\}$$

z_p - data vector; N_c - number of cluster centers;

m_j - j-th cluster center; C_{ij} - cluster centers

The fitness value of Fuzzy and Hybrid algorithms is obtained by

$$\sum_{i=}^{n}\sum_{j=1}^{c} u_{ij}^{m} d_{ij} \quad (9)$$

where u_{ij} - membership values; d_{ij} - distance; n – number of data points; c – number of clusters; m – fuzziness index

The Huang's accuracy measure [33] is determined by

Journal of Theoretical and Applied Information Technology
30th November 2014. Vol. 69 No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

$$r = \frac{\sum_{i=1}^{k} n_i}{n} \quad (10)$$

where n_i is the number of data occurring in both the i-th cluster and its corresponding true cluster, k is the number of clusters and *n* is the total number of data points in the data set. The accuracy rate (AR) and the error rate (ER) are determined by r x 100 and 100 – AR (11) respectively.

4.1.1 Hybrid FUZZY-cfPSO-FCM algorithm using kernel distance metric

The Kernel distance metric [34] [35] is defined as

$$d^2(x,z) = 2(1 - K(x,z)) \quad 12)$$

where $K(x,z) = \exp\frac{-\| x-z \|^2}{\sigma^2}$; (13)

$$\sigma = \sqrt{\frac{\sum_{i=1}^{n} u_{ij}{}^m \| x-z \|^2}{\sum_{i=1}^{n} u_{ij}{}^m}}; \quad (14)$$

$$u_{ij} = \frac{1}{\sum_{k=1}^{c}\left(\frac{d_{ij}^2}{d_{ik}^2}\right)^{\frac{1}{m-1}}}; 1 \le i \le n; 1 \le j \le c; \quad (15)$$

and $z_j = \frac{\sum_{i=1}^{n} u_{ij}^m K(x_i, z_j) x_i}{\sum_{i=1}^{n} u_{ij}^m K(x_i, z_j)}$ (16)

Algorithm Steps:

Input: Data set
Output: Cluster centers, fitness value, accuracy rate and error rate

Step 1: Choose the initial parameters - the number of cluster center c; select fuzziness index m (m>1); number of particles; initial velocity of particles; initial position of particles; pBest for each particle (p_{id}); gBest for the swarm (p_{gd}); acceleration constants c_1 and c_2 ,constriction factor χ , randomly generated random numbers r_1 and r_2; fix the maximum number of iterations *max_it*.

Step 2: Read the data set, randomly select initial fuzzy membership values $U^{(0)}$ for various particles.
Step 3: Find cluster centers for each particle.
Step 4: Compute the kernel distance d_{ij} for each particle using (12)
Step 5: For each particle update the fuzzy membership values $U^{(t+1)}$ using (15).
Step 6: Calculate the fitness value of each particle using (9)
Step 7: Compare every particle's fitness value with the previous particle's best solution and

$$p_{id}(t+1) = \begin{cases} p_{id}(t), & if\ f(x_{id}(t+1)) \ge f(p_{id}(t)) \\ x_{id}(t+1), & if\ f(x_{id}(t+1)) < f(p_{id}(t)) \end{cases} \quad (17)$$

Step 8: Compute the global best fitness value using

$$p_{gd}(t+1) = \arg\min_{p_i} f(p_{id}(t+1)), 1 \le i \le N \quad (18)$$

Step 9: Change the velocity of the particle according to

$$v_{id}(t+1) = \chi[v_{id}(t) + c_1 r_1 (p_{id}(t) - x_{id}(t)) + c_2 r_2 (p_{gd}(t) - x_{id}(t))] \quad (19)$$

Step 10: Change the position of the particle according to

$$x_{id}(t+1) = x_{id}(t) + v_{id}(t+1) \quad (20)$$

Step 11: Check the *max_it*, if not go to step 4.
Step 12: Record the final fuzzy membership values (U^{best}) and cluster center values (z^{best}).
Step 13: Select the number of clusters c; choose fuzziness index m (m>1); iteration error ε=0.00001; Fix the maximum number of iterations max_it; initialize the fuzzy membership values $U^{(0)} = U^{best}$ and clustering centers $z^{(0)} = \{z_1^{(0)}, z_2^{(0)}, ..., z_c^{(0)}\} = z^{best}$ for particle 1and randomly select the other particle membership values and compute the cluster centers.
Step 14: Calculate the kernel distance metric d_{ij} for each particle using (12)
Step 15: Update the new membership values of each particle using (15).
Step 16: Compute the fitness value of each particle using (9).
Step 17: Determine the particle best and global best (Best among various particles) and record the corresponding center values.
Step 18: Find the accuracy rate and error rate according to (11).
until the termination condition.

Journal of Theoretical and Applied Information Technology
30th November 2014. Vol. 69 No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

4.1.2 Hybrid FUZZY-cfPSO-FCM algorithm using mahalanobis distance metric

Mahalanobis distance [36] is the distance between a data point and center of a cluster, normalized by the standard deviation of the cluster in each dimension. It takes into account the membership as well as similarity between data point and cluster centroid. It is best suited model for clusters that have elliptical shapes. It is defined by

$$d(x,z)=\sqrt{(X-Z)^T\sigma^{-1}(X-Z)}=\sqrt{\sum_{i=1}^{d}\left(\frac{x_i-z_i}{\sigma_i}\right)^2} \quad (21)$$

where $x=(x_1,x_2,...,x_d)$ be a data point; $z=(z_1,z_2,...,z_d)$ be a center of a cluster; and σ_i be the standard deviation of points in the cluster in the i-th dimension. In the above hybrid algorithm 4.1.1, the distance metric is modified by using (21)

4.1.3 Hybrid FUZZY-cfPSO-FCM algorithm using new distance metric

A new distance metric was proposed by Kuo-Lung Wu et al. [14]. It is robust to noise and outliers and also handle unequal sized clusters. It is defined as

$$d^2(x,z)=1-\exp(-\beta\|x-z\|^2) \quad (22)$$

where $\beta=\left(\frac{\sum_{i=1}^{n}\|x_i-\bar{x}\|^2}{n}\right)^{-1}$ (23)

and $\bar{x}=\frac{\sum_{i=1}^{n}x_i}{n}$ (24)

In the above hybrid algorithm 4.1.1, the distance metric is replaced by using (22).

5. EXPERIMENTAL RESULTS

5.1 Data Sets

Five real-world UCI machine repository data sets [37], wine, vowel, liver disorders, glass and blood transfusion are applied to evaluate the performance of FCM, cfPSO and Hybrid algorithms.

The **Wine data set** is the results of chemical analysis of wines grown in the same region in Italy but derived from three different cultivars. The analysis determined the quantities of 13 constituents found in each of the three types of wines (class 1 – 59 instances; class 2 – 71 instances; class 3 – 48 instances). There are 178 objects with 13 numeric attributes – Alcohol; Malic acid; Ash; Alcalinity of ash; Magnesium; total phenols; Flavanoids; Nonflavanoid phenols; Proanthocyanins; Color intensity; Hue; OD280/OD315 of diluted wines and Proline. All the attributes are continuous.

The **Vowel data set** consists of 871 Indian Telugu vowel sounds. There are six vowel classes (class 1 – 72 patterns; class 2 – 89 patterns; class 3 – 172 patterns; class 4 – 151 patterns; class 5 – 207 patterns; class 6 – 180 patterns) and three input features. All entries are integers.

The **Liver Disorders data set** consists of 345 objects and two different types (class 1 – 145) objects; class 2 – 200 objects) characterized by six attributes including mcv, alkphos, sgpt, sgot, gammagt and drinks.

The **Glass data set** consists of 214 objects which are sampled from six different types of glasses: building windows float processed (70 objects), building windows nonfloat processed (76 objects), vehicle windows float processed (17 objects), containers (13 objects), table ware (9 objects) and headlamps (29 objects). Each type of glass has nine attributes: refractive index, sodium, magnesium, aluminum, silicon, potassium, calcium, barium and iron.

The **Blood Transfusion data set** consists of 748 donor data and 2 different types. All samples have 4 features. They are Recency – months since last donation (R), Frequency – total number of donation (F), Monetary – total blood donated in c.c. (M), and Time – months since first donation (T). The two class variables are not donating blood (class 1 – 570 instances) and donating blood (class 2 – 178 instances).

The above mentioned data sets are summarized in Table 1 .

Table 1: Description of Data Sets

Data set	No. of attributes	No. of Classes	Size
Wine	13	3	178
Vowel	3	6	871
Liver Disorders	6	2	345
Glass	9	6	214
Blood Transfusion	4	2	748

5.2 Results and Discussions

We have used a PC Pentium IV (CPU 3.06 GHZ and 1.97 GB RAM) for the experiments. The maximum number of iterations is 100 and each algorithm is tested through 10 independent runs. We have implemented the algorithms using Java.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

The selected parameters of FCM and cfPSO algorithms are given in Table 2.

Table 2: Description of Parameters

Name of the Parameter	Notation	Value
Cognitive component	c_1	2.05
Social component	c_2	2.05
Constriction factor	χ	0.729
No. of particles	N	10
Fuzziness index	M	2.0
Iteration error	ε	0.00001

The comparison of cluster center values of different algorithms on various data sets is shown in Table 3 to Table 7. The performance of the algorithms is evaluated through fitness value, accuracy rate and error rate. As shown in the Table 8 to Table 12, the hybrid FUZZY-cfPSO-FCM obtained minimum fitness value than other algorithms in all of data sets. The hybrid algorithm has the minimum fitness values of 11845.967, 208.707, 118.783 and 0.00029 on wine data set; 71546.404, 235.208, 290.698 and 0.000296 on vowel data set; 6266.618, 363.676, 345.024 and 0.03831 on liver disorders data set; 71.379, 89.328, 69.091 and 3.118 on glass data set and 297703.306, 612.487, 752.998 and 0.000063 on blood transfusion data set for Euclidean, Mahalanobis, Kernel and New distance metrics in each data set respectively. The accuracy and error rates of data sets wine, vowel, liver disorders, glass and blood transfusion are described in Table 13 to Table 17 respectively. The high accuracy (low error) rates are 98.88 (1.12); 97.1 (2.9) of wine, liver disorders data sets for Euclidean and Kernel respectively and 90.7 (9.3); 86.92 (13.08) and 99.06 (0.94) of vowel, glass and blood transfusion data sets for New distance metric. The comparison of fitness values of different algorithms on various data sets for Mahalanobis and Kernel distance metrics is shown in Figure 1 and Figure 2 respectively.

Table 3: Comparison of Center Values Produced by Different Algorithms on Wine Data Set

Distance	FCM	Cf-PSO	Hybrid
Euclidean	(12.992, 2.563, 2.390, 19.636, 104.027, 2.141, 1.636, 0.388, 1.529, 5.646, 0.891, 2.408, 742.706)	(13.396, 2.595, 2.608, 16.910, 111.244, 1.935, 4.095, 0.276, 2.883, 10.344, 1.358, 2.152, 697.842)	(13.803, 1.868, 2.457, 16.966, 105.355, 2.867, 3.027, 0.291, 1.921, 5.825, 1.081, 3.071, 1221.035)
	(12.515, 2.426, 2.295, 20.778, 92.423, 2.076, 1.788, 0.388, 1.454, 4.135, 0.946, 2.491, 459.580)	(11.699, 1.885, 2.346, 17.264, 115.739, 2.373, 3.267, 0.188, 1.276, 4.753, 0.642, 2.784, 1125.631)	(12.992, 2.563, 2.390, 19.636, 104.027, 2.141, 1.636, 0.388, 1.529, 5.646, 0.891, 2.408, 742.706)
	(13.803, 1.868, 2.457, 16.966, 105.355, 2.867, 3.027, 0.291, 1.921, 5.825, 1.081, 3.071, 1221.035)	(14.462, 4.428, 2.802, 20.684, 93.290, 2.562, 1.400, 0.627, 2.778, 9.970, 1.108, 2.543, 468.691)	(12.515, 2.426, 2.295, 20.778, 92.423, 2.076, 1.788, 0.388, 1.454, 4.135, 0.946, 2.491, 459.580)
Mahalanobosis	(12.453, 1.977, 2.268, 20.007, 93.735, 2.253, 2.059, 0.357, 1.584, 3.507, 1.030, 2.794, 568.792)	(13.517, 3.522, 2.711, 18.525, 93.026, 2.705, 1.965, 0.520, 2.750, 8.030, 1.014, 2.881, 1532.673)	(13.044, 3.198, 2.403, 21.014, 98.325, 1.762, 1.015, 0.444, 1.210, 6.577, 0.752, 1.890, 632.115)
	(13.580, 1.950, 2.423, 17.555, 105.624, 2.790, 2.867, 0.294, 1.903, 5.457, 1.053, 3.078, 1044.090)	(12.567, 2.493, 2.222, 20.929, 103.701, 2.419, 2.150, 0.288, 1.331, 4.397, 0.968, 2.650, 638.676)	(13.580, 1.950, 2.423, 17.556, 105.624, 2.790, 2.867, 0.294, 1.903, 5.457, 1.053, 3.078, 1044.090)
	(13.044, 3.198, 2.403, 21.014, 98.325, 1.762, 1.015, 0.444, 1.210, 6.577, 0.752, 1.890, 632.115)	(13.535, 1.999, 2.333, 16.686, 113.028, 2.379, 2.510, 0.341, 2.043, 6.999, 0.815, 3.154, 1158.570)	(12.453, 1.977, 2.268, 20.007, 93.735, 2.253, 2.059, 0.357, 1.584, 3.507, 1.030, 2.794, 568.792)
Kernel	(13.857, 1.831, 2.472, 16.959, 105.773, 2.913, 3.080, 0.292, 1.944, 6.101, 1.094, 3.043, 1291.828)	(12.706 ,5.077, 1.950, 24.275, 106.791, 3.414, 3.327, 0.341, 1.685, 8.302, 0.653, 3.088, 640.133)	(13.511, 1.973, 2.3633, 17.819, 2.716, 2.751, 0.293, 1.836, 4.9725, 1.048, 3.111, 993.311)
	(12.417, 2.267, 2.272, 20.664, 91.346, 2.234, 2.045, 0.373, 1.542, 3.585, 0.994, 2.667, 408.928)	(12.360, 0.740, 1.975, 18.694, 97.985, 1.236, 4.945, 0.564, 1.858, 5.199, 0.863, 1.907, 678.110)	(13.072, 2.636, 2.385, 19.257, 104.657, 2.201, 1.797, 0.380, 1.582, 5.466, 0.921, 2.501, 757.766)
	(12.837, 2.639, 2.381, 20.328, 99.111, 2.013, 1.421, 0.404, 1.410, 5.524, 0.867, 2.259, 656.794)	(13.874, 2.124, 2.212, 21.231, 144.686, 3.526, 3.863, 0.165, 1.544, 6.532, 1.185, 2.103, 522.575)	(12.712, 2.597, 2.355, 20.776, 94.883, 2.001,1.535,0.398, 1.382, 4.921, 0.893, 2.324, 581.338)
New	(13.724,1.953, 2.408, 16.998, 105.194, 2.805, 2.937, 0.285, 1.896, 5.391, 1.056, 3.130, 1107.725)	(13.199, 4.069, 1.919, 18.658, 121.151, 3.753, 2.184, 0.215, 1.955, 3.635, 0.798, 1.738, 695.564)	(13.040, 2.606, 2.392, 19.442, 105.149, 2.164, 1.691, 0.384, 1.557, 5.637, 0.899, 2.445, 761.235)
	(12.906, 2.598, 2.385, 19.906, 100.991, 2.063, 1.482, 0.405, 1.460, 5.748, 0.875, 2.280, 693.705)	(12.288, 1.796, 2.182, 21.466, 103.553, 1.863, 3.035, 0.331, 0.774, 12.026, 1.0931, 2.9038, 487.592)	(12.589, 2.490, 2.314, 20.781, 93.274, 2.027, 1.683, 0.393, 1.422, 4.461, 0.925, 2.412, 503.690)
	(12.502, 2.369, 2.294, 20.769, 92.163, 2.086, 1.826, 0.388, 1.462, 4.131, 0.955, 2.495, 452.952)	(11.227, 3.577, 1.656, 22.703, 84.705, 2.656, 3.615, 0.611, 1.302, 3.257, 0.963, 3.165, 1129.459)	(13.724, 1.929, 2.422, 17.026, 104.886, 2.817, 2.956, 0.289, 1.902, 5.490, 1.063, 3.104, 1128.893)

Journal of Theoretical and Applied Information Technology
30th November 2014. Vol. 69 No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 4: Comparison of Center Values Produced by Different Algorithms on Vowel Data Set

Distance	FCM	Cf-PSO	Hybrid
Euclidean	(409.940, 2093.966, 2653.469)	(465.113, 2218.329, 2758.026)	(409.944, 2093.958, 2653.464)
	(510.136, 1773.018, 2520.796)	(418.314, 994.414, 2363.703)	(361.167, 2293.167, 2970.200)
	(415.416, 1027.473, 2345.909)	(618.349, 1111.802, 2605.984)	(510.136, 1773.009, 2520.791)
	(442.610, 997.188, 2665.425)	(654.805, 1273.634, 2229.429)	(415.416, 1027.473, 2345.909)
	(644.406, 1290.701, 2298.109)	(442.191, 1476.337, 2452.059)	(644.407, 1290.699, 2298.108)
	(361.167, 2293.168, 2970.204)	(483.948, 1908.144, 2539.267)	(442.610, 997.188, 2665.425)
Mahalanobosis	(475.778, 1954.587, 2604.156)	(641.239, 2048.019, 1997.120)	(348.122, 2284.471, 2974.625)
	(544.340, 1259.467, 2632.583)	(440.301, 988.455, 2322.004)	(383.286, 2041.213, 2617.849)
	(694.838, 1308.578, 2271.458)	(721.308, 1395.240, 2246.414)	(690.230, 1297.626, 2282.514)
	(371.055, 979.342, 2590.150)	(416.773, 2191.334, 2931.983)	(430.557, 1088.933, 2333.919)
	(438.092, 1111.342, 2323.888)	(439.868, 1823.364, 2524.342)	(553.578, 1726.127, 2602.879)
	(346.052, 2265.706, 2924.367)	(466.610, 997.081, 2629.594)	(400.036, 988.379, 2626.536)
Kernel	(348.794, 2330.334, 3022.719)	(749.985, 1499.991, 2360.001)	(473.248, 1807.932, 2615.696)
	(492.635, 1890.714, 2558.986)	(372.571, 1413.925, 2337.182)	(494.717, 1314.162, 2486.472)
	(628.050, 1267.897, 2307.404)	(606.805, 2165.294, 2867.668)	(507.858, 1322.971, 2460.892)
	(405.114, 999.179, 2363.934)	(517.669, 1381.983, 2071.874)	(432.210, 1948.084, 2652.030)
	(386.769, 2166.849, 2711.509)	(549.713, 2363.661, 2818.886)	(451.294, 1180.330, 2569.835)
	(420.633, 953.799, 2672.764)	(591.868, 2024.759, 2610.780)	(491.038, 1648.905, 2543.905)
New	(508.562, 1725.860, 2493.794)	(394.706, 925.935, 2456.042)	(375.460, 2221.380, 2860.883)
	(437.863, 991.703, 2661.193)	(673.629, 700.000, 2712.416)	(462.722, 1967.566, 2610.105)
	(433.264, 2042.574, 2622.267)	(636.027, 1316.722, 2264.558)	(447.151, 1027.598, 2657.548)
	(415.560, 1025.605, 2340.471)	(436.828, 1989.355, 2600.186)	(625.182, 1305.663, 2332.938)
	(366.140, 2266.261, 2921.396)	(549.628, 1130.346, 2614.315)	(424.021, 1056.481, 2353.551)
	(653.844, 1278.041, 2289.901)	(434.557, 2550.000, 2887.458)	(512.624, 1680.067, 2496.315)

Table 5: Comparison of Center Values Produced by Different Algorithms on Liver Disorders Data Set

Distance	FCM	Cf-PSO	Hybrid
Euclidean	(91.398, 76.431, 52.375, 35.677, 105.809, 5.872)	(86.444, 111.754, 67.643, 8.063, 124.468, 14.612)	(89.839, 67.834, 25.251, 22.053, 24.268, 2.911)
	(89.839, 67.834, 25.251, 22.053, 24.268, 2.911)	(91.736, 67.131, 25.942, 22.333, 22.477, 4.589)	(91.398, 76.431, 52.375, 35.677, 105.809, 5.872)
Mahalanobosis	(89.461, 66.363, 25.168, 21.637, 26.829, 2.466)	(103.000, 138.000, 155.000, 82.000, 297.000, 20.000)	(89.461, 66.363, 25.168, 21.637, 26.829, 2.466)
	(91.433, 75.644, 39.766, 29.900, 58.395, 5.177)	(89.902, 68.731, 26.268, 21.849, 27.198, 983)	(91.433, 75.644, 39.766, 29.900, 58.395, 5.177)
Kernel	(89.763, 67.930, 24.962, 21.935, 23.218, 2.883)	(73.811, 47.879, 51.057, 25.769, 187.561, 2.381)	(90.443, 73.799, 35.19, 27.061, 49.941, 4.066)
	(93.956, 77.065, 63.469, 43.660, 178.039, 6.885)	(91.966, 54.096,21.987, 19.943, 5.000,1.324)	(89.980, 66.842, 26.668, 22.670, 29.345,3.021)
New	(89.627, 62.191, 22.030, 20.497, 18.456, 2.665)	(97.643, 23.000, 4.000, 29.898, 130.477, 15.441)	(90.711, 75.220, 40.293, 29.541, 63.683, 4.607)
	(90.537, 76.516, 30.955, 24.729, 37.515, 3.509)	(87.075, 64.719, 27.704, 20.065, 24.361, 7.060)	(89.934, 67.093, 25.824, 22.300, 26.777, 2.928)

Table 6: Comparison of Center Values Produced by Different Algorithms on Blood Transfusions Data Set

Distance	FCM	Cf-PSO	Hybrid
Euclidean	(10.217, 3.266, 816.462, 27.434)	(74.000, 1.000, 2228.242, 53.501)	(7.008, 14.228, 3557.037, 60.866)
	(7.008, 14.228, 3557.037, 60.866)	(41.520, 28.811, 522.240, 12.664)	(10.217, 3.2656, 816.462, 27.434)
Mahalanobosis	(9.524, 2.990, 747.447, 21.151)	(8.588, 4.358, 767.461, 25.743)	(8.646, 10.376, 2594.094, 57.888)
	(8.646, 10.376, 2594.094, 57.888)	(13.761, 7.374, 2102.469, 98.000)	(9.524, 2.990, 747.447, 21.151)
Kernel	(5.104, 38.186, 9546.529, 89.124)	(0.160, 50.000, 2530.446, 98.000)	(10.362, 3.374, 843.427, 27.679)
	(10.036, 3.726, 931.611, 28.884)	(0.0, 1.000, 250.000, 2.000)	(7.698, 9.435, 2358.666, 48.604)
New	(10.955, 2.125, 531.288, 22.187)	(21.768, 30.809, 10698.324, 47.469)	(7.113, 11.405, 2851.181, 54.157)
	(7.925, 7.699, 1924.763, 45.902)	(16.929, 1.000, 746.467, 34.117)	(10.254, 3.360, 840.000, 27.586)

Journal of Theoretical and Applied Information Technology
30th November 2014. Vol. 69 No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 7: Comparison of Center Values Produced by Different Algorithms on Glass Data Set

Distance	FCM	Cf-PSO	Hybrid
Euclidean	(1.517, 13.333, 3.514, 1.406, 72.635, 0.577, 8.298, 0.025, 0.049)	(1.511, 14.721, 1.693, 1.778, 73.728, 5.331, 10.071, 2.809, 0.325)	(1.527, 11.974, 0.077, 1.080, 72.168, 0.250, 14.120, 0.119, 0.088)
	(1.517, 12.938, 3.435, 1.358, 72.966, 0.593, 8.474, 0.021, 0.068)	(1.511, 11.678, 3.335, 2.949, 71.826, 3.641, 15.308, 2.460, 0.205)	(1.517, 14.610, 0.076, 2.207, 73.203, 0.083, 8.694, 1.052, 0.019)
	(1.520, 13.488, 0.409, 1.526, 72.864, 0.368, 11.076, 0.064, 0.054)	(1.511, 14.508, 1.089, 1.982, 71.778, 1.099, 10.408, 0.609, 0.285)	(1.520, 13.488, 0.409, 1.526, 72.864, 0.368, 11.076, 0.064, 0.054)
	(1.517, 14.609, 0.076, 2.207, 73.203, 0.083, 8.694, 1.052, 0.019)	(1.511, 16.163, 2.572, 2.053, 71.053, 4.068, 8.438, 1.749, 0.181)	(1.517, 13.333, 3.514, 1.406, 72.635, 0.577, 8.298, 0.025, 0.049)
	(1.521, 13.772, 3.538, 0.945, 71.876, 0.218, 9.480, 0.040, 0.058)	(1.511, 14.963, 2.824, 2.250, 75.122, 5.166 10.149, 1.856, 0.460)	(1.521, 13.772, 3.538, 0.945, 71.876, 0.218, 9.480, 0.040, 0.058)
	(1.527, 11.974, 0.077, 1.080, 72.168, 0.250, 14.120, 0.119, 0.088)	(1.511, 13.054, 3.496, 1.297, 72.639, 0.539, 8.533, 0.168, 0.431)	(1.517, 12.938, 3.435, 1.358, 72.966, 0.593, 8.474, 0.021, 0.068)
Mahalanobosis	(1.517, 12.993, 3.299, 1.364, 72.850, 0.595, 8.609, 0.047, 0.202)	(1.511, 11.137, 4.490, 0.328, 71.107, 6.126, 7.079, 2.740, 0.000)	(1.522, 13.374, 0.791, 1.437, 72.407, 0.386, 11.322, 0.099, 0.044)
	(1.517, 13.035, 3.398,1.384, 72.927, 0.591, 8.436, 0.021, 0.012)	(1.511, 12.214, 3.630, 0.290, 75.410, 6.210, 10.479, 0.125, 0.510)	(1.517, 14.558, 0.140, 2.214, 73.153, 0.100, 8.743, 1.015, 0.019)
	(1.522, 13.374, 0.791, 1.437, 72.407, 0.386, 11.322, 0.099, 0.044)	(1.511, 10.730, 2.907, 0.812, 70.624, 0.000, 7.514, 0.000, 0.510)	(1.518, 13.345, 3.337, 1.405, 72.599, 0.561, 8.513, 0.039, 0.024)
	(1.517, 14.558, 0.140, 2.214, 73.153, 0.100, 8.743, 1.015, 0.019)	(1.511, 10.730, 4.490, 3.496, 72.670, 0.000, 9.560, 0.000, 0.199)	(1.522, 13.741, 3.429, 0.932, 71.873, 0.207, 9.649, 0.048, 0.040)
	(1.522, 13.741, 3.429, 0.932, 71.873, 0.207, 9.649, 0.048, 0.040)	(1.511, 14.401, 1.061, 0.290, 75.076, 0.000, 13.734, 3.082, 0.510)	(1.517, 12.993, 3.299, 1.364, 72.850, 0.595, 8.609, 0.047, 0.202)
	(1.518, 13.345, 3.337, 1.405, 72.599, 0.561, 8.513, 0.039, 0.024)	(1.511, 13.416, 2.495, 1.523, 72.760, 0.505, 8.528, 0.075, 0.000)	(1.517, 13.035, 3.398, 1.384, 72.927, 0.591, 8.436, 0.021, 0.012)
Kernel	(1.527, 11.987, 0.0549, 1.067, 72.112, 0.289, 14.140, 0.144, 0.084)	(1.511, 13.438, 3.309, 1.675, 72.411, 0.315, 8.575, 0.000, 0.000)	(1.518, 13.428, 2.690, 1.448, 72.680, 0.492, 8.899, 0.185, 0.055)
	(1.522, 13.778, 3.611, 0.854, 71.781, 0.223, 9.596, 0.032, 0.057)	(1.511, 10.730, 2.443, 0.795, 74.315, 2.186, 9.560, 2.094, 0.000)	(1.519, 13.334, 2.656, 1.441, 72.637, 0.486, 9.066, 0.207, 0.057)
	(1.517, 14.626, 0.0345, 2.181, 73.197, 0.122, 8.639, 1.122, 0.020)	(1.511, 10.730, 1.697, 1.621, 74.081, 2.247, 8.126, 3.150, 0.499)	(1.518, 13.417, 2.712, 1.449, 72.562, 0.612, 8.900, 0.149, 0.051)
	(1.517, 13.326, 3.497, 1.404, 72.651, 0.581, 8.296, 0.038, 0.052)	(1.511, 17.380, 2.692, 1.254, 75.410, 3.183, 16.190, 3.150, 0.313)	(1.518, 13.422, 2.705, 1.459, 72.691, 0.505, 8.894, 0.187, 0.076)
	(1.517, 12.929, 3.432, 1.383, 72.967, 0.638, 8.399, 0.035, 0.065)	(1.511, 13.996, 3.381, 0.988, 72.621, 3.430, 10.257, 1.059, 0.429)	(1.518, 13.347, 2.6748, 1.404, 72.659, 0.689, 8.977, 0.078, 0.0505)
	(1.521, 13.415, 0.298, 1.487, 72.892, 0.340, 11.305, 0.061, 0.054)	(1.511, 15.403, 2.722, 2.177, 74.254, 0.622, 11.755, 1.096, 0.166)	(1.519, 13.411, 2.704, 1.506, 72.523, 0.705, 8.941, 0.130, 0.050)
New	(1.517, 13.317, 3.571, 1.342, 72.657, 0.550, 8.343, 0.007, 0.048)	(1.511, 17.380, 3.731, 2.036, 73.636, 2.102, 11.251, 2.317, 0.338)	(1.517, 14.427, 0.203, 2.210, 73.109, 0.123, 8.988, 0.857, 0.024)
	(1.517, 14.653, 0.021, 2.235, 73.238, 0.020, 8.738, 1.028, 0.020)	(1.511, 11.972, 1.574, 2.218, 75.410, 3.245, 10.093, 3.150, 0.210)	(1.520, 13.566, 1.266, 1.569, 72.495, 0.497, 10.172, 0.247, 0.063)
	(1.519, 13.471, 3.368, 1.333, 72.235, 0.497, 8.894, 0.021, 0.043)	(1.511,12.690, 1.828, 2.568, 69.810, 1.499, 10.692, 3.150, 0.278)	(1.517, 12.963, 3.403, 1.353, 72.939, 0.585, 8.519, 0.026, 0.064)
	(1.517, 13.058, 3.512, 1.498, 72.952, 0.599, 8.141, 0.006, 0.055)	(1.511, 12.775, 1.165, 1.765, 71.885, 3.233, 12.859, 0.735, 0.205)	(1.520, 13.602, 1.920, 1.534, 72.380, 0.463, 9.722, 0.217, 0.082)
	(1.517, 12.848, 3.474, 1.296, 73.005, 0.583, 8.586, 0.009, 0.059)	(1.511, 16.252, 1.911, 2.629, 72.742, 0.000, 9.933, 0.687, 0.226)	(1.517, 13.283, 3.446, 1.379, 72.669, 0.565, 8.421, 0.027, 0.049)
	(1.522, 13.879, 3.712, 0.760, 71.763, 0.123, 9.641, 0.006, 0.050)	(1.511, 13.194, 3.944, 1.469, 72.997, 0.516, 8.565, 0.000, 0.300)	(1.521, 13.602, 3.269, 1.083, 72.075, 0.318, 9.433, 0.074, 0.059)

Table 8: Comparison of Fitness Values Produced by Different Algorithms on Wine Data Set

Distance	FCM	Cf-PSO	Hybrid
Euclidean	12196.410	16487.750	**11845.967**
Mahalanobosis	229.253	600.617	**208.707**
Kernel	270.887	356.000	**118.783**
New	0.0003	0.0005	**0.00029**

Table 9: Comparison of Fitness Values Produced by Different Algorithms on Vowel Data Set

Distance	FCM	Cf-PSO	Hybrid
Euclidean	76700.870	170979.801	**71546.404**
Mahalanobosis	281.436	746.624	**235.208**
Kernel	899.848	1742.000	**290.698**
New	0.000313	0.0009	**0.000296**

Journal of Theoretical and Applied Information Technology
30th November 2014. Vol. 69 No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 10: Comparison of Fitness Values Produced by Different Algorithms on Liver Disorders Data Set

Distance	FCM	Cf-PSO	Hybrid
Euclidean	7687.794	10852.291	**6266.618**
Mahalanobosis	392.508	723.296	**363.676**
Kernel	591.855	690.000	**345.024**
New	0.03949	0.0771	**0.03831**

Table 11: Comparison of Fitness Values Produced by Different Algorithms on Glass Data Set

Distance	FCM	Cf-PSO	Hybrid
Euclidean	88.429	355.119	**71.379**
Mahalanobosis	108.992	756.856	**89.328**
Kernel	206.072	411.938	**69.091**
New	3.191	17.628	**3.118**

Table 12: Comparison of Fitness Values Produced by Different Algorithms on Blood Transfusion Data Set

Distance	FCM	Cf-PSO	Hybrid
Euclidean	354036.975	413938.341	**297703.306**
Mahalanobosis	677.851	1102.010	**612.487**
Kernel	1391.675	1496.000	**752.998**
New	0.000067	0.0001	**0.000063**

Table 13: *Comparison of Accuracy and Error Rate Produced by FCM and Hybrid Algorithms on Wine Data Set*

Distance	**Criteria**	**FCM**	**Hybrid**
Euclidean	Accuracy Rate (%)	98.88	**98.88**
	Error Rate (%)	1.12	**1.12**
Mahalanobosis	Accuracy Rate (%)	94.94	96.63
	Error Rate (%)	5.06	3.37
Kernel	Accuracy Rate (%)	68.54	83.71
	Error Rate (%)	31.46	16.29
New	Accuracy Rate (%)	89.89	93.82
	Error Rate (%)	10.11	6.18

Table 14: *Comparison of Accuracy and Error Rate Produced by FCM and Hybrid Algorithms on Vowel Data Set*

Distance	**Criteria**	**FCM**	**Hybrid**
Euclidean	Accuracy Rate (%)	86.8	90.59
	Error Rate (%)	13.2	9.41
Mahalanobosis	Accuracy Rate (%)	81.17	87.94
	Error Rate (%)	18.83	12.06
Kernel	Accuracy Rate (%)	76.23	80.02
	Error Rate (%)	23.77	19.98
New	Accuracy Rate (%)	84.73	**90.7**
	Error Rate (%)	15.27	**9.3**

Table 15: *Comparison of Accuracy and Error Rate Produced by FCM and Hybrid Algorithms on Liver Disorders Data Set*

Distance	**Criteria**	**FCM**	**Hybrid**
Euclidean	Accuracy Rate (%)	73.33	73.33
	Error Rate (%)	26.67	26.67
Mahalanobosis	Accuracy Rate (%)	71.59	87.54
	Error Rate (%)	28.41	12.46
Kernel	Accuracy Rate (%)	66.67	**97.1**
	Error Rate (%)	33.33	**2.9**
New	Accuracy Rate (%)	83.77	83.77
	Error Rate (%)	16.23	16.23

Table 16: *Comparison of Accuracy and Error Rate Produced by FCM and Hybrid Algorithms on Glass Data Set*

Distance	**Criteria**	**FCM**	**Hybrid**
Euclidean	Accuracy Rate (%)	80.37	82.71
	Error Rate (%)	19.63	17.29
Mahalanobosis	Accuracy Rate (%)	74.3	78.5
	Error Rate (%)	25.7	21.5
Kernel	Accuracy Rate (%)	63.08	65.42
	Error Rate (%)	36.92	34.58
New	Accuracy Rate (%)	71.03	**86.92**
	Error Rate (%)	28.97	**13.08**

Table 17: *Comparison of Accuracy and Error Rate Produced by FCM and Hybrid Algorithms on Blood Transfusion Data Set*

Distance	**Criteria**	**FCM**	**Hybrid**
Euclidean	Accuracy Rate (%)	95.05	95.05
	Error Rate (%)	4.95	4.95
Mahalanobosis	Accuracy Rate (%)	92.25	92.25
	Error Rate (%)	7.75	7.75
Kernel	Accuracy Rate (%)	79.01	95.32
	Error Rate (%)	20.99	4.68
New	Accuracy Rate (%)	80.35	**99.06**
	Error Rate (%)	19.65	**0.94**

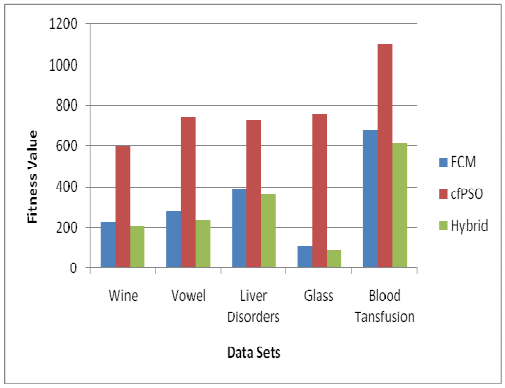

Figure 1: Comparison of Fitness Value for Mahalanobis Distance Metric

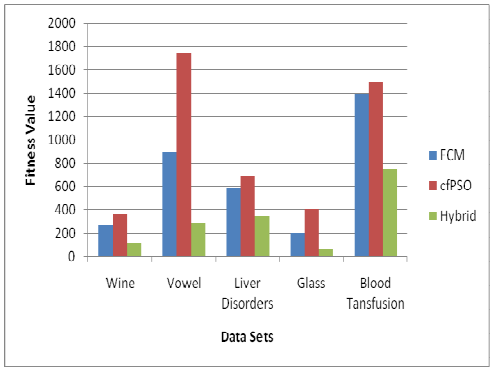

Figure 2: Comparison of Fitness Value for Kernel Distance Metric

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

6. CONCLUSIONS

Data Clustering is the task of grouping a set of data points in such a way that the data points in the same cluster are more similar to each other than to those in other clusters. Fuzzy c-means (FCM) algorithm is easily trapped in local optimal values. The results of FCM algorithm are highly depended on initial membership values. Constriction Factor Particle Swarm Optimization (cfPSO) algorithm is a popular swarm intelligence technique and applied to solve the data clustering problems. Euclidean distance metric is commonly used in most data clustering algorithms. It is sensitive to the scales of the variables involved. It is suitable to model the spherical shapes. It is not appropriate for elliptical shapes or for handling outlier data points. In this paper, FUZZY-cfPSO-FCM algorithm is presented using Non-Euclidean Distance Metrics such as Kernel, Mahalanobis and New Distance. The center values of FCM, cfPSO and Hybrid algorithm are recorded over five real-world data sets, wine, glass, vowel, liver disorders and blood transfusion. The results are also evaluated through the fitness value, accuracy rate and error rate. Experimental results show that proposed hybrid method is efficient in terms of minimum fitness value, error rate and high accuracy rate.

REFRENCES:

[1] J. Han and M. Kamber, "Data mining: concepts and techniques", Morgan Kaufmann, San Francisco, 2001.

[2] J. MacQueen, "Some methods for classification and analysis of multivariate observations", *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, Vol. 1, 1967, pp. 281-297.

[3] L. Kaufman and P. Russeeuw, "Finding groups in data: an introduction to cluster analysis", New York, John Wiley & Sons, 1990.

[4] J.C. Dunn, "A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters", *Journal of Cybernetics*, Vol. 3, 1973, pp. 32-57.

[5] J.C. Bezdek, "Pattern recognition with fuzzy objective function algorithms", Plenum Press, New York, 1981.

[6] J. Kennedy and R. Eberhart, "Particle Swarm Optimization", *IEEE International Conference on Neural Networks*, Piscataway, NJ, Vol. 4, 1995, pp. 1942-1948.

[7] R.C. Eberhart and Y. Shi, "Comparing inertia weights and constriction factors in particle swarm optimization", *IEEE Proceedings of the Congress Evolutionary Computation*, 2000, pp. 84-88.

[8] M. Clerc, J. Kennedy, "The particle swarm – Explosion, stability, and convergence in a multi-dimensional complex space", *IEEE Trans. Evol. Comput.,* Vol. 6, 2002, pp. 58-73.

[9] R.J. Hathway and J. Bezdek, "Optimization of clustering criteria by reformulation", *IEEE Transactions on Fuzzy System*, 1995, pp. 241-245.

[10] J. Kennedy and R. Eberhart, "Swarm Intelligence", Morgan Kaufmann Publishers, Inc., San Francisco, CA, 2001.

[11] K. Hirota and W. Pedrycz, "Fuzzy computing for data mining", *Proceedings of the IEEE*, Vol. 87(9), 1999, pp. 1575-1600.

[12] LiXiang Jun, Wu You Xin, Qiu Ji Guang and Wan Li Hui, " The applications of fuzzy c-means clustering in macro-economic forecast", *Second International Symposium on Electronic Commerce and Security*, Vol. I, 2009, pp. 609-611.

[13] Wuling Ren, Jinzhu Cao and Xianjie Wu, "Application of network intrusion detection based on fuzzy c-means clustering algorithm", *Third International Symposium on Intelligent Information Technology Application*", Vol. 3, 2009, pp. 19-22.

[14] Kuo-Lung Wu, and Miin-Shen Yang, "Alternative C-means Clustering Algorithms", *Pattern Recognition*, Vol. 35, 2002, pp. 2267-2278.

[15] D.Q. Zhang and S.C. Chen, "A novel kernel-based fuzzy and possibilistic c-means clustering", *Proceedings of the International Conference on Artificial Neural Networks*, Istanbul, Turkey, 2003, pp. 122-125.

[16] Bighnaraj Naik and Sarita Mahapatra, "Cooperative Swarm Intelligence Based Evolutionary Approach to find optimal cluster center in cluster analysis", *Journal of Theoretical and Applied Information Technology*, Vol. 42, No. 1, 2012, pp. 8-17.

[17] M. Omran, A. Salman and A. P. Engelbrecht, "Image classification using particle swarm optimization", *Proceedings of the4th Asia-Pacific Conference on Simulated Evolution and Learning,* Singapore, 2002, pp. 370-374.

[18] D.W. Van Der Merwe and A.P. Engelbrecht, "Data clustering using particle swarm optimization", *Proceedings of the IEEE Congress on Evolutionary Computation*, Canberra, Australia, 2003, pp. 215-220.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[19] A.A.A. Esmin, D.L. Pereira and F. de Araujo, "Study of different approach to clustering data by using the particle swarm optimization algorithm", *Proceedings of the IEEE World Congress on Evolutionary Computation*, 2008, pp. 1817-1822.

[20] O.A. Mohamed Jafar and R. Sivakumar, "A Study of Bio-inspired Algorithm to Data Clustering Using Different Distance Measures", *International Journal of Computer Applications (IJCA)*", Vol. 66, No. 12, 2013, pp. 33-44.

[21] Y. Kao and S.Y. Lee, "Combining K-means and particle swarm optimization for dynamic data clustering problems", *Proceedings of the IEEE International Conference on Intelligent Computing and Intelligent System*, 2009, pp. 757-761.

[22] Fun Ye and Ching-Yi Chen, "Alternative KPSO-Clustering Algorithm", Tamkang *Journal of Science and Engineering*, Vol. 8, No. 2, 2005, pp. 165-174.

[23] X. Cui and T.E. Potok, "Document Clustering Analysis Based on Hybrid PSO+K-means Algorithm", *Journal of Computer Science*, 2005, pp. 27-33.

[24] T. Niknam and b. Amiri, "An efficient hybrid approach based on PSO, ACO and K-means for cluster analysis", *Applied Soft Computing*, Vol. 10, No. 1, 2010, pp. 183-197.

[25] Mehdi Neshat, Shima Farshchian Yazdi, Daneyal Yazdani and Mehdi Sargolzaei, "A New Cooperative Algorithm based on PSO and K-Means for Data Clustering", *Journal of Computer Science*, 8(2), 2012, pp. 188-194.

[26] Li Yi Ran, Zhu Yong Yong and Zhang Chun Na, "The K-means Clustering Algorithm based on Chaos Particle Swarm", *Journal of Theoretical and Applied Information Technology*, Vol. 48, No. 2, 2013, pp. 762-767.

[27] W. Pang, K. Wang, C. Zhou and L. Dong, "Fuzzy discrete particle swarm optimization for solving traveling salesman problem", *Proceedings of the fourth international conference on computer and information technology*, 2004, pp.796-800.

[28] T.A. Runkler and C. Katz, "Fuzzy Clustering by Particle Swarm Optimization", *IEEE International Conference on Fuzzy Systems*, Canada, 2006, pp. 601-608.

[29] E. Mehdizadeh, S. Sadi-Nezhad and R. Tavakkoli-Moghaddam, "Optimization of Fuzzy Clustering Criteria by a Hybrid PSO and Fuzzy C-Means Clustering Algorithm", *Iranian Journal of Fuzzy Systems*, Vol. 5, No. 3, 2008, pp. 1-14.

[30] H. Izakian and A. Abraham, "Fuzzy C-means and fuzzy swarm for fuzzy clustering problem", *Expert Systems with Applications*, 38, 2011, pp. 1835-1838.

[31] Yong Zhang, Dan Huang, Min Ji and Fuding Xie, "Image segmentation using PSO and PCM with Mahalanobis distance", *Expert Systems with Applications*, 38, 2011, pp. 9036-9040.

[32] Chaoshun Li, Jianzhong Zhou, Pangao Kou and Jian Xiao, "A novel chaotic particle swarm optimization based fuzzy clustering problem", *Neurocomputing*, 83, 2012, pp. 98-109.

[33] Z. Huang, and M.K. Ng, "A fuzzy k-modes algorithm for clustering categorical data", *IEEE Trans. Fuzzy Systems*, 7(4), 1999, pp. 446-452.

[34] B. Scholkopf, "The Kernel Trick for Distances", In: Advances in Neural Information Processing Systems 13, *Fourteenth Annual Neural Information Processing Systems Conferences (NIPS 2000)*, MIT Press, Cambridge, MA, USA, 2001, pp. 301-307.

[35] M. Gimiami, "Mercer kernel based clustering in feature space", *IEEE Trans. on Neural Networks*, 13(3), 2002, pp. 780-784.

[36] R.D. Maesschalck, D. Jouan-Rimbaud, and D. Massart, "The mahalanobis distance", *Chemometrics and Intelligent Laboratory Systems*, Vol. 50, No. 1, 2000, pp. 1-18.

[37] UCI repository of machine learning databases, University of California-Irvine, Department of Information and Computer Science, http://www.ics.uci.edu/~mlearn/MLRepository.html.