Volume 3, Issue 7, July 2013 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Particle Swarm Optimization Algorithm Based *k*-means and Fuzzy *c*-means clustering

Asha Gowda Karegowda *
Department of Master of Computer Applications, Siddaganga Institute of Technology, Tumkur, India

Seema Kumari
Department of Master of Computer Applications Siddaganga Institute of Technology, Tumkur, India

***Abstract*—Data mining is the process of extracting hidden patterns from huge data. Among the various clustering algorithms, k-means is the one of most widely used clustering technique in data mining. The performance of k-means clustering depends on the initial clusters and might converge to local optimum. K-means does not guarantee the unique clustering because it generates different results with randomly chosen initial clusters for different runs of k-means. In addition, the performance of fuzzy c-means depends on membership matrix [μ] value and might not guarantee the unique clustering. This paper explains the application of evolutionary algorithm namely Particle Swarm Optimization and Entropy based fuzzy clustering for identifying the initial centroids for enhancing the performance of both k-means and fuzzy c-means clustering.**

***Keywords*—k-means clustering, fuzzy c-means clustering, cluster initialization, Particle swarm optimization, Entropy based fuzzy clustering**

I. INTRODUCTION

The data mining functionalities mainly include association rule mining, classification, prediction & clustering. Clustering is the process of grouping the data into the classes or clusters so that the objects within a cluster have similarity in comparison to one another, but are very dissimilar to the objects in other cluster [1]. One of most common clustering method is a k-means clustering. The main drawback of the k-means algorithm is that the result is sensitive to the selection of the initial cluster centroids and may converge to the local optima. Therefore, the initial selection of the cluster centroids effects the main processing of the K-means. However, if good initial clustering centroids can be obtained using evolutionary algorithm like PSO, GA, ABC, the *k*-means would work well in refining the clustering centroids to find the optimal clustering centres [2]. In this paper, PSO and Entropy based fuzzy clustering (EFC) algorithm have been used to identify the optimal centroids for both k-means and fuzzy k-means clustering.

II. K-MEANS CLUSTERING

K-means [3,4] is one of the simplest unsupervised learning algorithms and follows partitioning method for clustering. *K*-means algorithm takes the input parameter, *k* as number of clusters and partitions a dataset of n objects into *k* clusters, so that the resulting objects of one cluster are dissimilar to that of other cluster and similar to objects of the same cluster. In k-means algorithms begins with randomly selected *k* objects, representing the k initial cluster center or mean. Next each object is assigned to one the cluster based on the closeness of the object with cluster center. To assign the object to the closest center, a proximity measure namely Euclidean distance is used that quantifies the notion of closest. After all the objects are distributed to *k* clusters, the new *k* cluster centers are found by taking the mean of objects of *k* clusters respectively. The process is repeated till there is no change in *k* cluster centers. *K*-means algorithm aims at minimizing an objective function namely sum of squared error (SSE). SSE is defined as

$$E = \sum_{i=1}^{k} \sum_{p \in C_i} |p - m_i|^2 \quad \text{- (1)}$$

where *E* is sum of the square error of objects with the cluster means for k cluster. p is the object belong to a cluster C_i and m_i is the mean of cluster C_i .The time complexity of *k*-means is O (t*k*n) where *t* is the number of iterations, *k* is number of clusters and *n* is the total number of records in dataset.

III. FUZZY C-MEANS CLUSTERING

Fuzzy *c*-means [5,6] is an extension of *k*-means clustering. The major difference between the fuzzy *c*-means and *k*-means is that the later discovers hard clusters where a particular sample can belong to only one cluster while the former discovers soft clusters where a particular sample can belong to more than one cluster with certain probability. This belongingness of a data sample to the cluster is represented using membership values.

Let there be *N* data points each of *L* dimensions. The object X_{ij} represents i^{th} object with j^{th} dimension with *i = 1 to N ; j = 1* to L. The working of fuzzy c-means algorithm is discussed using following steps.

i. Input the number of clusters and appropriate level of cluster fuzziness $g > 1$. Initialize the $N \times C$ sized membership matrix $[\mu]$ at random, such that $\mu_{ij} \in [0.0, 1.0]$ and

$$\sum_{j=1}^{k} \mu_{ij} = 1 \quad (2)$$

for i = 1 to N, j= 1 to k.

ii. Calculate k^{th} dimension of the j^{th} cluster center CC_{jk} using the equation (3)

$$CC_{jk} = \frac{\sum_{i=1}^{N} \mu_{ij}^{g} x_{ik}}{\sum_{i=1}^{N} \mu_{ij}^{g}} \quad - (3)$$

iii. Compute the Euclidean distance d_{ij} , between i^{th} sample and j^{th} cluster center.

iv. Update fuzzy membership matrix $[\mu]$ according to d_{ij}. If $d_{ij} >= 0$, then

$$\mu = \frac{1}{\sum_{m=1}^{C} \left(\frac{d_{ij}}{d_{im}}\right)^{\frac{2}{g-1}}} \quad - (4)$$

If $d_{ij} = 0$ then the sample coincides with j^{th} cluster center CC^{j} and it will have full membership value that is $\mu_{ij} = 1$.

v. Repeat steps ii-iv until the change in matrix $[\mu]$ is less than some user specified value.

vi. The data points are grouped based on their similarity of membership values.

In the fuzzy c-means clustering the membership matrix $[\mu]$ is initialized randomly in step 1, which in turn is used to compute the centroids. Further the new centroids are used to compute the matrix $[\mu]$ by finding the Euclidean distance between the samples and the centroids. The process is repeated till there is user specified minimum change in the matrix $[\mu]$. Thus the performance of fuzzy *c*-means depends on the randomly initialized matrix $[\mu]$. This paper proposes modified fuzzy c-means clustering algorithm by first initializing the centroids using 3 different methods: Random, *EFC*, and PSO methods. The Euclidean distance between the samples and the initial centroids identified by the proposed method is used to compute membership matrix $[\mu]$. The matrix $[\mu]$ in turn is used to compute new centroids and the process is repeated till user specified minimum change in the matrix $[\mu]$ is achieved. Experimental results show a remarkable improvement of fuzzy *c*-means clustering performance using the proposed method compared to traditional fuzzy *c*-means clustering.

IV. PARTICLE SWARM OPTIMIZATION

PSO is a population-based stochastic search algorithm. It was first introduced by Kennedy and Eberhart. Since then, it has been widely used to solve a broad range of optimization problems. It attempts to mimic the natural process of group communication to share individual knowledge when such swarms flock, migrate, or hunt. If one member sees a desirable path to go, the rest of this swarm will follow quickly. In PSO, this behaviour of animals is imitated by particles with certain positions and velocities in a searching space, wherein the population is called a swarm, and each member of the swarm is called a particle. Starting with a randomly initialized population, each particle in PSO flies through the searching space and remembers the best position it has seen. Members of a swarm communicate good positions to each other and dynamically adjust their own position and velocity based on these good positions [9,10]. The velocity adjustment is based upon the local best of each particle and global best of the swarm using equation 5. In this way, the particles tend to fly towards better and better searching areas over the searching process. The searching procedure based on this concept can be described as:

$$v_{ij}(t + 1) = wv_{ij}(t) + {}_{c1R1}(pbest_{ij} - x_{ij}(t)) + c_2R_2(gbest_{ij} - x_{ij}(t)) \quad - (5)$$

Where, v_i is the velocity of particle x_i is the current position of particle *w* is the weighting function, *c1*& *c2* are the constants (usually 2) which determine the relative influence of the social and cognitive components, p_i is the pbest of particle i, p_g is the gbest of the group[6]. The weighting function is computed using equation

$$w = W\max - \frac{W\max - W\min}{Iter(\max)} - Iter(x) \quad - (6)$$

where, *wmax* is the initial weight, *wmin* is the final weight, itermax is the maximum iteration number, *iter* is the : current iteration number[11].

Using the above equation, a certain velocity, which gradually gets close to *pbest* and *gbest* can be calculated. The current position (searching point in the solution space) can be modified by the following equation.

$$x_{ij}(t + 1) = x_{ij}(t) + v_{ij}(t + 1) \quad -(7)$$

V. WORKING OF PSO BASED K-MEANS AND FUZZY C-MEANS CLUSTERING

i. Initialization: The swarm population is randomly generated, where each particle represents the cluster centroids. Dimension of each particle is taken as product of dataset dimension and number of clusters to be generated.

ii. The fitness of each particle is computed by the following fitness function
Objective function is sum of square error (SSE) for all clusters computed by equation
$\sum||x^i - z^j||,\ i=1,..,n,\ j=1,..,c$
where n and c are the number of samples and number of clusters, respectively and || || is the Euclidean distance between the sample x^i and cluster centroid z^j, represented by each particle.

iii. Update the local best and global best position of each particle using SSE.
Update the particle position, i.e. cluster centroids using (5) and (7)

iv. Repeat steps ii-iv till the maximum iterations is reached.

The global best particle represents the best initial centroids for the both k-means and fuzzy c-means clustering.

VI. ENTROPY BASED FUZZY CLUSTERING (EFC)

Yao introduced EFC [13], which identifies the number of clusters and initial cluster prototypes by itself. The entropy is calculated for each sample using equation (8).

$$E_i = \sum_{k \in x}^{j \neq i} (S_{ij} \log_2 S_{ij} + (1 - S_{ij}) \log_2 (1 - S_{ij})) \quad (8)$$

where $S_{ij} = e^{-\alpha d_{ij}}$ is the similarity between two data points (*i, j)* and d_{ij} is the Euclidean distance between points (*i, j)*

The algorithm for entropy based fuzzy clustering is as follows. The inputs for the algorithm are dataset *D* with *N* samples, β the threshold value, can be viewed as a threshold of similarity among the data points in the same cluster, an constant α is which is computed as (ln 0.5/ ($\bar{D}$)), where $\bar{D}$ is the mean distance among the pairs of data points in a hyper-space and is usually set to 0.5.

Step 1. Compute entropy E_i for each sample x_i from dataset *D for I = 1 to N*

Step 2. Identify x_i that has the minimum E_i value as the cluster centre.

Step 3. Remove x_i and data points having similarity x_i greater than some threshold β from *D*

Step 4. If *D* is not empty then go to step 2.

The k centroids identified by EFC, are selected as k-means and fuzzy c-means initial cluster centres.

VII. RESULTS

Experiments have been conducted using several datasets availed from publicly available UCI machine learning datasets namely Diabetes, wine, sonar, iris, ionosphere and heart-statlog. The PSO and EFC have been used to identify the initial centroids for both *k*-means and fuzzy *c*-means clustering. For PSO, experiments have conducted by varying the population size and number of iterations. With EFC the experiments have been conducted by varying the value of value of threshold β. For fuzzy *c*-means clustering, the fuzzifier value *g* has been varied in the range of 1.2 to 2.0. The clustering result for fuzzy c-means clustering was found to be best with 2.0. The clustering accuracy of k-means clustering with PSO identified centroids proved to be always better when compared to that of EFC and random centroids. The clustering accuracy of fuzzy k-means clustering was found be same with PSO and EFC identified centroids and better when compared to those of random centroids. It was observed that the EFC identified centroids gave better results with more number of iterations when compared to PSO centroids. The comparative results of *k*-means and fuzzy *c*-means clustering with random, PSO and EFC initialized centroids is shown in Figure 1 and Figure 2 respectively. The PSO and EFC identified centroids enhanced the fuzzy c-means clustering performance by an order of 14.63%, 12.50%, 24.72%, 10.54%, 2.65% and 26.33% for heart stat log, diabetes, wine, ionosphere, sonar and iris dataset respectively. Similarly the PSO identified centroids enhanced the k-means clustering performance by an order of 6%, 6.18%, 8.83%, 4.51%, 16.55% and 9.16% for heart stat log, diabetes, wine, ionosphere, sonar and iris dataset respectively. EFC identified centroids showed a lower performance when compared with PSO identified centroids for almost all 6 datasets.

VIII. CONCLUSIONS

The performance of *k*-means and fuzzy *c*-means clustering depends on the random selected centroids and randomly initialized membership matrix respectively. This paper illustrates the applications of PSO to identify the initial centroids for improving the accuracy of both *k*-means and fuzzy *c*-means clustering. Experiments have been conducted on six different datasets garnered from the UCI machine learning datasets. The results of PSO identified centroids proved to be better when compared with EFC centroids and random method for *k*-means clustering. However, PSO and EFC resulted in same clustering accuracy for Fuzzy *c*-means clustering for all the six datasets.

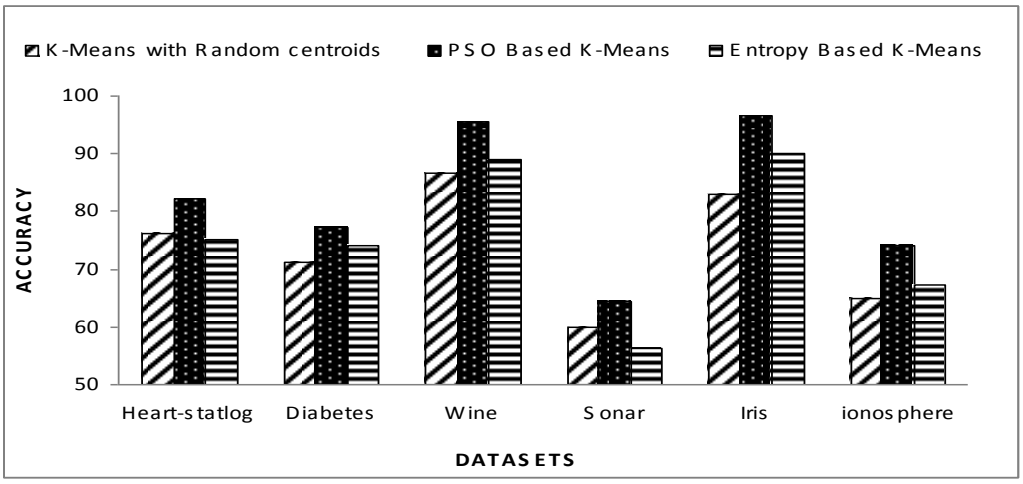

Figure 1. Performance measures of k-means clustering by initializing cluster centers using random, PSO and EFC identified centroids for various datasets

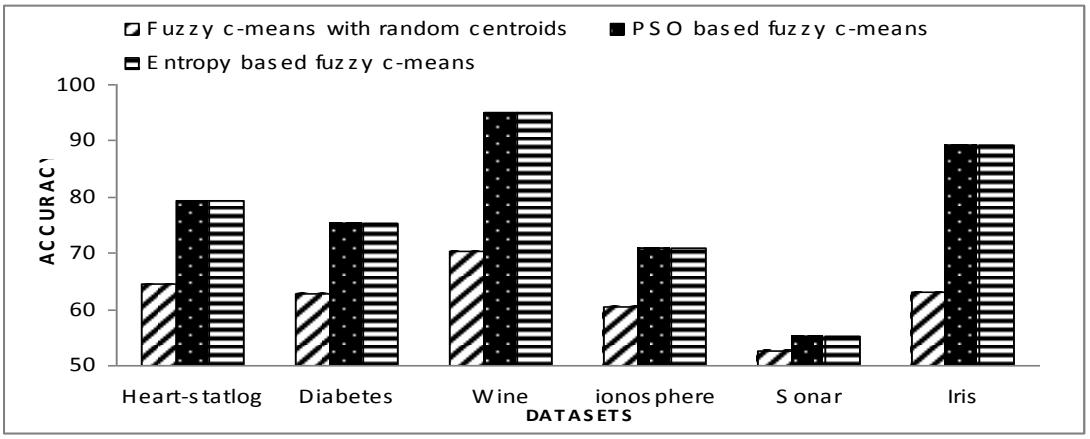

Figure 2. Performance measures of fuzzy *c*-means clustering by initializing cluster centers using random, PSO and EFC identified centroids for various datasets

REFERENCES

[1] Han, and M.Kamber, Data Mining: Concepts and techniques, San Francisco, Morgan Kauffmann Publishers, 2001.
[2] Hartigan, J. A. "Clustering Algorithms, John Wiley and Sons, Inc., New York, 1975.
[3] Anderberg, M. R., Cluster Analysis for Applications. Academic Press, Inc., New York, 1973.
[4] Mac Queen J, " Some methods for the classification and analysis of multivariate observations ", Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probility, Berkeley, University of California Press, pp 281-297, 1967.
[5] Dunn J.C. "A fuzzy relative of the ISODATA process and its used in detecting compact well-separated clusters. Journal of Cybernetics", vol. 3, pp. 32-57, 1973.
[6] Bezdek J.C, "Fuzzy mathematics in pattern classification", Ph.D thesis, Applied Mathematics Center,Ithca:Cornell University, 1973.
[7] Pradeep Rai, Shubha Singh, "A survey of clustering techniques", international journal of computer application, vol. 7, No. 12, 2010.
[8] DW van der Merwe, AP Engelhrecht, "Data Clustering using Particle Swarm Optimization", proceeding of IEEE congress on Evolutionary computation, vol. 1 pp. 215-220, 2003.
[9] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory", Proceedings of the Sixth International Symposium on Micro Machine and Human Science, Nagoya, Japan, pp.39-43, 1995.
[10] R. Eberhart and Y. Shi, "Comparing inertia weights and constriction factors in particle swarm optimization", Proc. of Congress on Evolutionary Computation (CEC2000), San Diego, CA, pp 84-88, 2000.
[11] Suresh Chandra Satapathy, Gunanidhi Pattnaik,etal, 'Performance comparisons of PSO based Clustering', international journal of Computer Science and Networking, Vol. 1, issue 1, December 2009.
[12] Sandeep Rana, Sanjay Jasola, Rajesh Kumar, " Hybrid sequential approach for data clustering using K-means and PSO algorithm", International journal of Engineering, Science and Technology, vol. 2,No. 6, pp 167-176, 2010.
[13] Yao J, M. Dash, S T Tan, Liu H., "Entropy based fuzzy clustering and fuzzy modelling", Fuzzy Sets and Systems, vol. 113, pp 381-388, 2000.

