World Applied Sciences Journal 10 (5): 574-583, 2010
ISSN 1818-4952

An Efficient Fuzzy K-Medoids Method

[1]Moh'd Belal Al-Zoubi, [2]Ali Al-Dahoud and [1]Mousa AL-Akhras

[1]Computer Information Systems Department,
King Abdullah II School for Information Technology,
The University of Jordan, P.O. Box 13835, Amman 11942, Jordan
[2]Computer Science Department, Faculty of Science, Al- Zaytoonah University, Amman, Jordan

Abstract: In this paper, we propose an efficient fuzzy k-medoids clustering method (will be termed FKM). The fuzzy c-means clustering algorithm is first executed producing the membership grade matrix. The medoids are then selected as the objects with the highest membership grades in each cluster. Different datasets have been tested and the results showed that the proposed method is faster than the well-known Partitioning Around Medoids (PAM) method in all cases. When datasets of high dimensionality were applied or when there is large number of data points in the dataset, our proposed method took a significantly less computational time and the difference in CPU running time is more apparent. In terms of the quality of partitioning, the proposed FKM method and the PAM method perform almost the same with marginal differences that we show to be statistically insignificance.

Key words: Data Mining · Clustering · Fuzzy Clustering · K-Medoids · Partitioning Around Medoids

INTRODUCTION

Clustering plays an important role in a wide variety of fields including data mining, artificial intelligence, pattern recognition and medical sciences. The goal of clustering is to discover the natural groups (or clusters) of objects in datasets and hence to examine similarities and dissimilarities among objects [1].

Existing clustering methods can be classified into two main categories: hierarchical and partitioning methods [2,3]. Hierarchical methods find nested clusters recursively either in agglomerative or divisive modes. Partitioning methods attempt to partition the objects in the dataset into *k* groups or clusters, by optimizing a criterion function, which is usually defined over all objects in a dataset.

Hierarchical methods in agglomerative mode given *n* objects to be clustered begin with *n* clusters; i.e. put each object in a cluster of its own. In each step, two clusters are chosen and merged. This process continues until all objects are clustered into one group. Divisive methods begin by putting all objects in one cluster. In each step, a cluster is chosen and split up into two clusters. This process continues until n clusters are produced.

Hierarchical methods are known to suffer from many points of weakness as reported by Ng and Han [2], weaknesses include the fact that that they can never undo what was done previously in the agglomerative or divisive modes. It has also been reported that the clusters produced by a partitioning method are of higher quality than the clusters produced by a hierarchical method [1, 2].

Therefore, developing partitioning methods has been one of the main focuses of cluster analysis research and many partitioning methods have been developed based on k-means, fuzzy clustering methods or k-medoids. These methods attempt to find a "good" partitioning in the sense that objects in the same cluster are close or related to each other, while objects of different clusters are far or different from each other [1].

The k-means algorithm starts with a random initial partition and keeps reassigning the objects to clusters based on the similarity between the objects and the cluster centers until there is no reassignment of any object from one cluster to another [1, 4].

Unlike the k-means partition algorithm, in which each object in the dataset can belong to only one cluster,

Corresponiding Author: Moh'd Belal Al-Zoubi, Computer Information Systems Department King Abdullah II School for Information Technology, The University of Jordan, P.O. Box 13835, Amman 11942, Jordan.
Tel: + 962 (0) 6 53 55 000 extension 22602, E-mail: mba@ju.edu.jo & aldahoud@alzaytoonah.edu.jo.

fuzzy clustering methods attempt to generate a partition in which each object in the dataset can belong to every cluster with a certain membership degree, using a membership function [5].

K-medoid methods are partitioning clustering methods that partition a dataset of n objects into *k* groups or clusters. *K* objects are selected, which are centrally located (or optimal representative objects) in the constructed clusters, called medoids and the clustering method is called k-medoid method.

Various k-medoids methods have been published [6]. Most of these algorithms are very time-consuming [7]. In addition, they have deficiencies in handling high dimensional datasets [8].

The cluster centers produced by the k-means algorithm or fuzzy clustering methods are just representatives of the clusters. These centers are placed anywhere in the space, even where it does not make sense (for example, to place a hospital in the middle of a street), while the k-medoids models require that medoids are restricted to a subset of the original dataset. For this reason k-medoids method was chosen for improvement in this paper

In this paper, we propose a new fast fuzzy k-medoids method that can handle high dimensional datasets efficiently. The proposed method will be compared to the Partitioning Around Medoids (PAM) method [9, 10], as one of the most widely used k-medoids methods. For its accuracy, PAM is considered as a benchmarked k-medoids clustering method by many researchers [6, 8-10]. The results produced by our proposed method will be compared to those produced by the PAM method in terms of time efficiency and accuracy, using different benchmarked datasets.

Related Work: PAM is one of the most effective and widely used k-medoids methods [6, 8-10]. PAM starts by selecting an initial set of medoids and iteratively replaces each one of the selected medoids by one of the non-selected medoids in the dataset as long as the sum of dissimilarities between the objects and their closest medoids is improved. The process is repeated until the criterion function converges.

The major drawback of the PAM algorithm is that the time required for partitioning the data is very expensive and can become unmanageable for large datasets [10, 11]. With n objects and *k* clusters, the time complexity of the PAM method is estimated as $O(k(n - k)^2)$ [12].

Other partitioning methods based on PAM were published, such as Clustering Large Applications (CLARA) [10] and Clustering Large Applications based on Randomized Search (CLARANS) [1].

CLARA creates multiple samples (of $40 + 2k$) objects randomly from the entire dataset and then applies PAM to each of the samples. The CLARA method, however, requires multiple scans of the entire dataset, which ends up in an extra CPU time. The time complexity of each iteration of the algorithm is estimated as $O(ks2 + k(n-k))$, where *s* is the size of the sample [13]. It has been reported that the produced clustering solution may not be as good as the solution produced by the whole set [14].

The CLARANS method stems from the work done on PAM and CLARA. It relies on a randomized search of a graph to find the medoids which represent the clusters. However, it has been reported that the CLARANS method has a high time complexity, estimated as $O(n^2)$ for each iteration [7, 13]. In addition, it has been reported in [8] that the deficiency of the method is that it does not function well when high dimensional data is applied.

It should be noted that PAM, CLARA and CLARANS generate crisp clusters. When the clusters are not well defined (i.e. when they overlap) fuzzy clusters are desired [15].

In this paper, we propose a Fuzzy k-Medoid (FKM) method based on the Fuzzy c-Means (FCM) clustering algorithm [16, 17]. The FKM method produces quality results similar to the standard PAM method with less time complexity when applied to two dimensional datasets. More noticeable reduction in time complexity is achieved using the FKM method when applied to datasets with high dimensionality in comparison to the time complexity of the PAM method.

Proposed Method: Our proposed FKM method is based on the FCM clustering algorithm [16, 17]. In the proposed method, the FCM algorithm is executed first, producing the membership degree matrix. Medoids are then selected as the objects with the highest membership grade values in each cluster.

The main merit of the FCM clustering algorithm is that its speed of convergence is fast and little storage space is needed [18, 19]. The time complexity of the FCM algorithm is estimated as $O(nk^2)$ for each iteration [12], where *n* is the number of objects and *k* is the number of fuzzy clusters. Comparing between the time complexity of the FCM and the PAM method, a significant increase in

the time efficiency can be achieved by using FCM especially when applied to high dimensional datasets. The advantages of the FCM include a straightforward implementation, fairly robust behavior, applicability to multidimensional data and the ability to model uncertainty within the data [1, 20].

FCM partitions a collection of n data points x_i, $i = 1, \ldots, n$ into k fuzzy clusters and finds a cluster center in each cluster such that the objective function of the dissimilarity measure is minimized. FCM employs fuzzy partitioning such that a given data point can belong to several clusters with the degree of belongingness specified by membership grades between *0* and *1*. The membership matrix U is allowed to have elements with values between *0* and *1* and the summation of degrees of belongingness for a dataset should always be equal *to 1*, as shown in the constraint:

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

The goal of the FCM is to find the fuzzy cluster centers (centroids) and the fuzzy membership grades minimizing the following objective function:

$$J = \sum_{i=1}^{n} \sum_{j=1}^{k} u_{ij}^{m} d_{ij}^{2} \quad (2)$$

Where u_{ij}values are between *0* and *1*; c js the cluster center of fuzzy cluster j; $d_{ij} = \left\| x_i - c_j \right\|$ is the Euclidean distance between I^{th} data point *and* j cluster and $m \in [1, \infty)$ is a weighting exponent that controls the degree of fuzziness of the clusters found by minimizing the objective function in equation (2).

The necessary conditions for equation (2) to reach its minimum are:

$$c_j = \frac{\sum_{i=1}^{n} u_{ij}^{m} x_i}{\sum_{i=1}^{n} u_{ij}^{m}} \quad (3)$$

and

$$u_{ij} = \frac{1}{\sum_{q=1}^{k} \left(\frac{d_{ij}}{d_{qi}} \right)^{2/(m-1)}} \quad (4)$$

The FCM algorithm determines the cluster centers c_j and the membership matrix U using the following steps [21]:

Step [1]: Initialize the membership matrix U with random values between 0 and 1 such that the constraints in equation (1) are satisfied.

Step [2]: Calculate k fuzzy cluster centers c_j , $j = 1, \ldots, k$, using equation (3).

Step [3]: Compute the objective function according to equation (2). Stop if either it is below a certain tolerance value or its improvement over previous iteration is below a certain threshold, ε.

Step [4]: Compute a new U using equation (4). Go to step 2.

The cluster centers (centroids) produced by the FCM are just representatives of the clusters. These centers are placed anywhere in the space, even where it does not make sense (for example, to place a school on a street), while the k-medoids models require that medoids are restricted to a subset of the original set. Therefore, our aim is to find "real" objects (medoids) among the dataset to represent each cluster. This can be achieved by exploiting the values of the U membership grade matrix produced by the FCM algorithm. These values represent the membership grade between each object and different clusters. The higher the membership grade is, the stronger an object belongs (relates) to its cluster as the most centrally located object in the cluster and hence the most likely to be the medoid of the cluster.

To show how the proposed FKM method works, we will use the dataset presented by Chianga and Yin [22] as an example. The dataset consists of 20 data points, with two dimensions, as shown in Figure 1. It is clear that the dataset contains two "natural" clusters.

Applying the FCM algorithm to the points of the dataset produces membership grade matrix shown in Table 1.

The table shows the point's ID (ID) in the first column. The second column shows the membership grades (U_1) for the data points regarding cluster 1 and the third column shows the membership grades (U_2) for the data points regarding cluster 2.

The table shows that the point with the highest (maximum) value in the second column is the point with ID = 5 (bold). The point with the highest value in the third column is the point with ID = 15 (bold). Hence, our two medoids are the points with ID =5 and ID = 15.

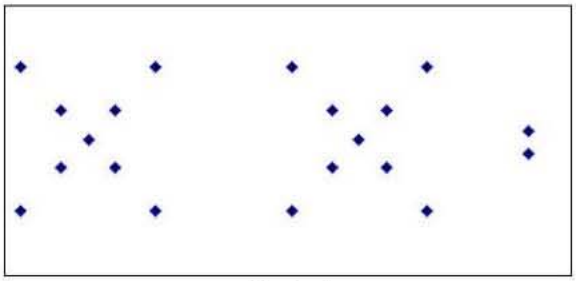


Fig. 1: Data set presented in [22]

Table 1: The membership grades for the data set shown in Figure 1

ID	U1	U2
1	0.93812	0.06188
2	0.93821	0.06179
3	0.98581	0.01419
4	0.98583	0.01417
5	**0.99993**	6.56E-05
6	0.98323	0.01677
7	0.98326	0.01674
8	0.87398	0.12602
9	0.87434	0.12566
10	0.25183	0.74817
11	0.25407	0.74593
12	0.07214	0.92786
13	0.07318	0.92682
14	0.01696	0.98304
15	0.00861	**0.99139**
16	0.00941	0.99059
17	0.04509	0.95491
18	0.04648	0.95352
19	0.08597	0.91403
20	0.08640	0.91360

RESULTS AND DISCUSSION

In order to measure the performance of our proposed FKM method, we have compared our results to the results obtained from applying the PAM method when different datasets are applied. All of the experiments have been implemented on Pentium IV / 1GH personal computers, using the MATLAB software package (2007a, The MathWorks, Inc. Natick, MA, USA).

As discussed in [7], there is no universally agreed upon definition of the quality of the clustering results. The majority of researchers describe a cluster by considering the internal homogeneity and the external separation; i.e. objects in the same cluster are related to each other, while objects of different clusters are far or different from each other [1].

Several techniques have been proposed for measuring clustering quality results. One technique is the Average Silhouette width, another technique is the use of the Average Distance (AD) between each object and its nearest medoid. This technique has been proposed by Kaufman and Rousseeuw [10] and will be used in this research. The technique allows two solutions to be compared for a given dataset: the smaller the value of the AD is, the better the solution is.

We will start our experimentations with the Ruspini dataset obtained from [10]. This dataset has 75 objects (data points) and two dimensions as shown in Figure 2.

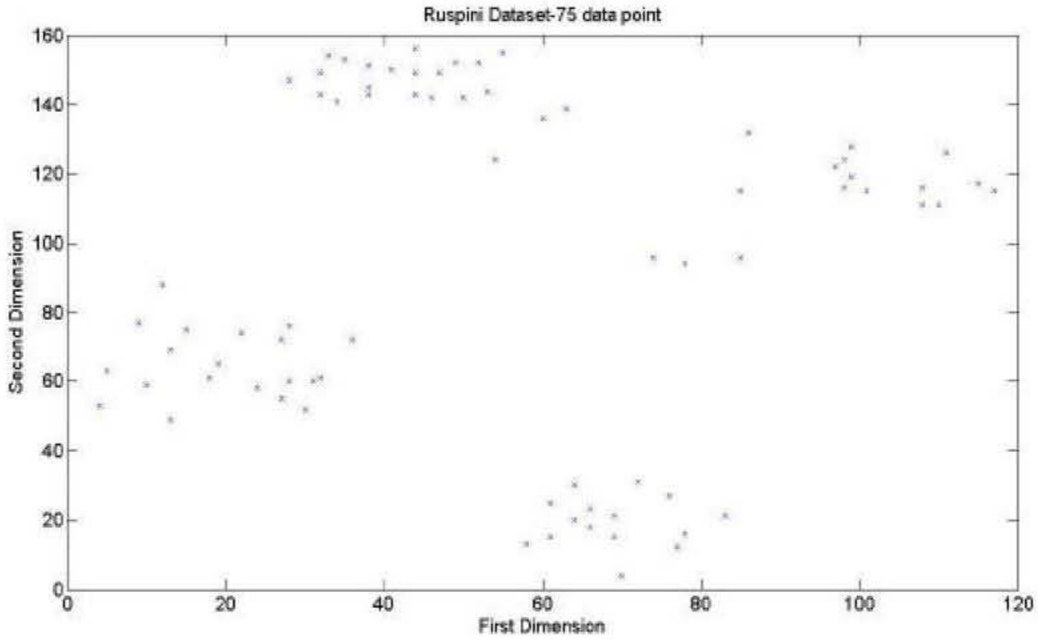
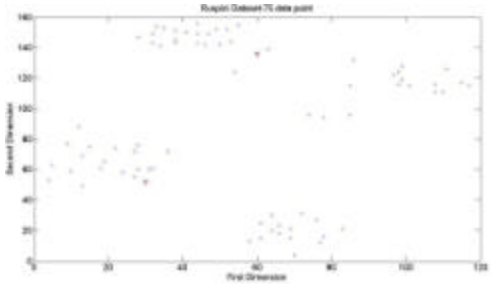
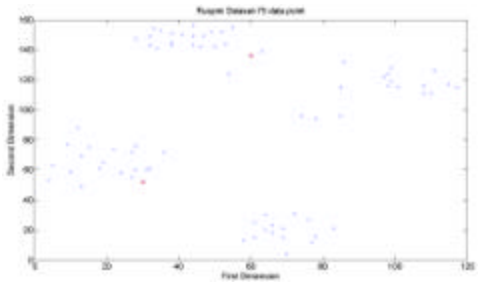


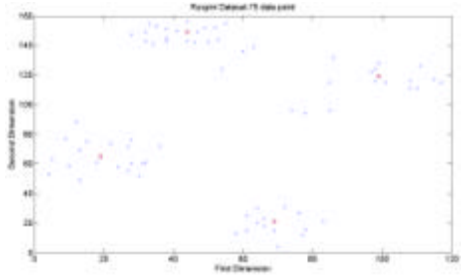
Fig. 2: Ruspini data points



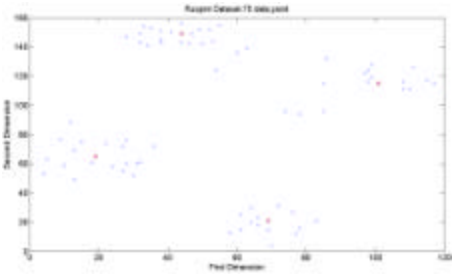
PAM clustering, K=2



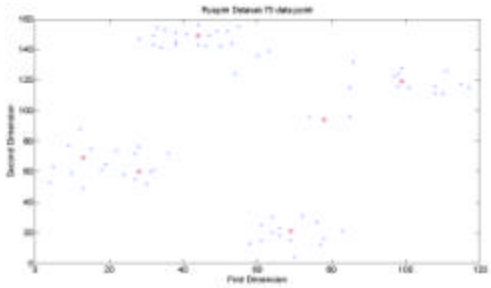
FKM clustering, K=2



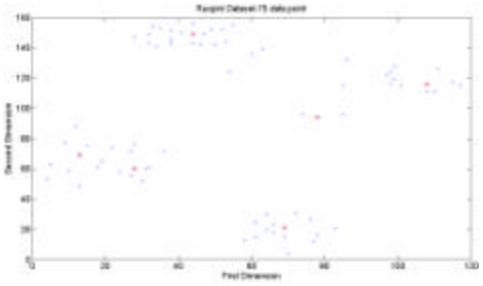
PAM clustering, K=4



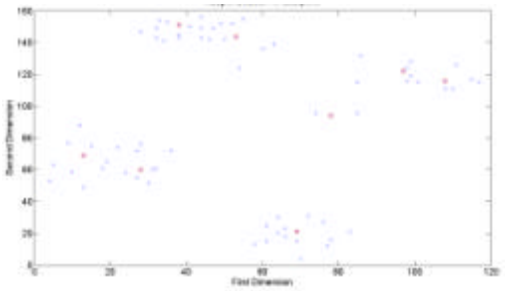
FKM clustering, K=4



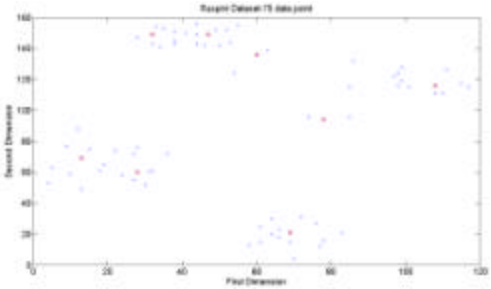
PAM clustering, K=6



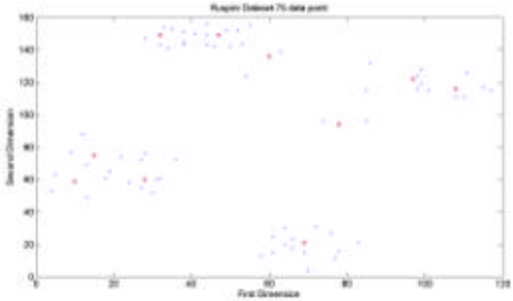
FKM clustering, K=6



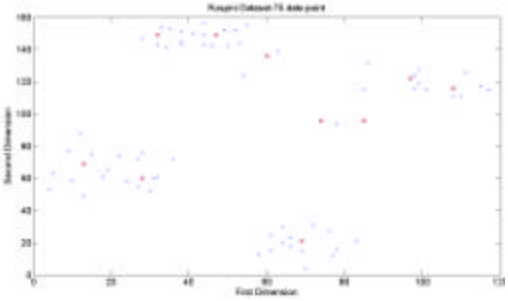
PAM clustering, K=8



FKM clustering, K=8



PAM clustering, K=10



FKM clustering, K=10

Fig. 3: Ruspini dataset clustering using PAM and FKM, K=2, 4, 6, 8, and 10

Table 2: The AD values produced by the PAM and the FKM methods when the Ruspini dataset is applied, K=2, 4, 6, 8 and 10.

K	PAM	FKM	Difference	t-value
2	31.94	31.94	0.00	NA
4	11.48	11.49	0.01	0.0006
6	9.53	9.68	0.15	0.0052
8	8.00	8.25	0.25	0.0344
10	6.84	7.34	0.50	0.0221

Different number of clusters are attempted, specifically k=2, 4, 6, 8 and 10. Figure 3 shows the resulted medoids from the PAM method and from the proposed FKM method. The resulted medoids are marked in filled circles.

Table 2 shows the AD values produced by the PAM method and the proposed FKM method. The first column shows the number of medoids (clusters), *k*, the second column represents the AD values obtained from applying the PAM method and the third column represents the AD values obtained from the FKM method. The fourth column computes the differences between the AD values obtained from the FKM methods minus the AD values obtained from the PAM method. The fifth column computes the t-values to test if the differences are statistically significant [23]. The degree of freedom is set to 74 as there are 75 data points.

The results of Table 2 show that PAM gave better results in most cases, although the FKM method gave the same results as PAM when k = 2. It can be noticed from Table 2 that the differences in the results obtained from the PAM and the FKM methods are marginal.

From the t-values shown in the fifth column of Table 2, it can be concluded that there is no significant differences between the values produced by the PAM method and the values produced by the FKM method at the 1% significance level (99% confidence level) for all values of k as all the t-values are included in the interval [-2.5758, 2.5758] which confirms the effectiveness of the proposed method in finding the best medoids [23].

When K=2, both methods performed the same as the AD values from both methods were the same. This is because both methods found the same medoids (Points 17 and 42) with coordinates (30, 52) and (60, 136), respectively. When K=4, the proposed FKM method has very similar performance to that of PAM. The difference in AD is 0.01 with t-value of 0.0006 which is not statistically significant at the 99% confidence level [23].

Although the PAM method achieved slightly better results than the FKM method with k values greater than 4, looking at the distribution of the data points shown in Figure 2, it can be noticed that the data points are distributed ideally into four clusters, which indicates that using more clusters is not advantageous in both methods. The rest of the results in the table are self-explanatory.

Figure 4 shows the box plots of differences between the data points of the Ruspini dataset and the corresponding medoids produced using the PAM method and the proposed FKM method [24].

When k=2, the differences achieved by the PAM and the FKM method are the same as shown in Figure 4 (a) and (b), respectively. This is because both methods found the same medoids. From Figure 4 (c) it can be noticed that when PAM method is used with k=4, 25% of the differences (the first quartile) are below 7.05. The lower two quartiles (median value) have maximum difference of 10.44. The maximum difference for the third quartile is 14.2. The maximum difference for the majority of points is 24.04 with few outliers of higher differences. With the same number of cluster k=4, when FKM method is used as shown in Figure 4 (d), points of the first quartile are below 7.21. The median has the value of 10.3. The maximum difference for the third quartile is 14.2. The maximum difference is 24.04 with more outliers than in Figure 4 (c) which justifies the marginal difference between the two methods.

When k=6 and PAM method is used in Figure 4 (e), the first quartile data points have differences below 5.099. The median has the value of 9.22. The maximum difference for the third quartile is 13.26. The maximum difference for the majority of points is 21.47. When FKM is used in Figure 4 (f), 25% of the differences (the first quartile) are below 5.873. The median has the value of 9.22. The maximum difference for the third quartile is 12.53. The maximum difference for the majority of points is 22.14. Also, in this case more outliers exist than the case of PAM which justifies the differences between the two methods. Similar results and justifications exist in Figure 4 (g), (h), (i) and (j) with PAM (K=8), FKM (K=8), PAM (K=10) and FKM (K=10), respectively.

Table 3 shows the CPU run time in milliseconds for both methods, when a different number of clusters, *k*, is used. The table shows that in all cases the FKM method is very efficient in terms of the CPU time. The last column shows that in all cases, the improvement (speedup) which is achieved by the FKM method is significant.

In the rest of this section, we will investigate the effectiveness of the FKM method when high dimensional datasets are used. High dimensional clustering is required for many applications including Geographic Information Systems (GIS) and gene expression data applications.

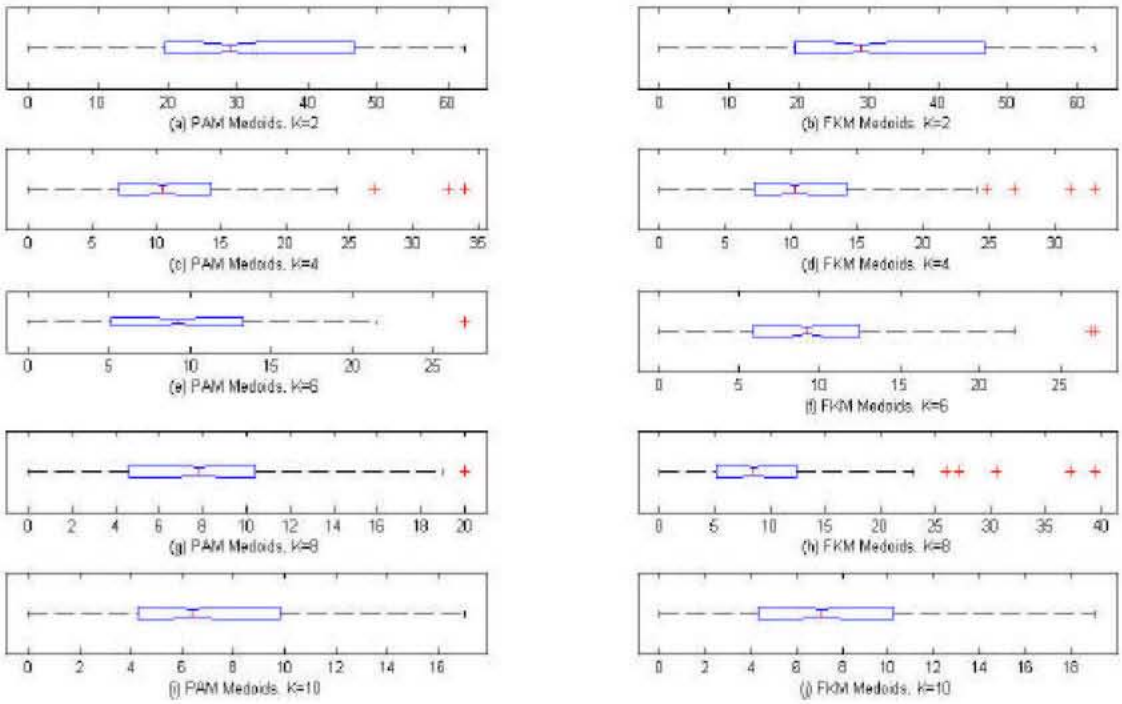


Fig. 4: Box plot of differences between the data points of the Ruspini datasets and the medoids produced using PAM and FKM methods. K=2, 4, 6, 8 and 10

Table 3: CPU run time in milliseconds for Ruspini data.

K	PAM	FKM	Improvement
2	13	6	54%
4	32	16	50%
6	47	31	34%
8	93	45	52%
10	140	78	44%

Table 4: Summary of the datasets obtained from the University of California, Irvine repository of machine learning databases: n is the number of objects (instances), d is the dimensions of the set and k is the number of classes (clusters) [25].

Data set	n	d	k
Iris	150	4	3
Wood	20	6	2
Bupa	345	6	2
Yeast	1484	8	3

It has been reported in [8] that most of the existing k-medoids methods discussed above (i.e. PAM, CLARA and CLARANS) do not function well when high dimensional data is applied.

To test the effectiveness of the FKM method when high dimensional datasets are applied, different benchmarked datasets whose true classes (clusters) are known have been used. These datasets are obtained from the University of California, Irvine repository of machine learning databases [25] and are summarized in Table 4.

Table 5: The AD values produced by the PAM and the FKM methods for different high dimensional datasets

Data set	PAM	FKM	Difference	t-value
Iris	6.54	6.63	0.09	0.0051
Wood	0.1348	0.1348	0	NA
Bupa	29.2056	29.2056	0	NA
Yeast	0.2174	0.2216	0.0042	-0.0029

Table 5 shows the AD values produced by the PAM method and the proposed FKM method. The first column shows the name of the dataset, the second column represents the AD values obtained from applying the PAM method and the third column represents the AD values obtained from the FKM method. The fourth column computes the differences between the AD values obtained from the FKM methods minus the AD values obtained from the PAM method. The fifth column computes the t-values to test if the differences are statistically significant. The degree of freedom in each case is set to the number of data points in that dataset minus 1 [23].

The results of Table 5 show that PAM gave slightly better results than FKM method for Iris dataset although the difference is statistically insignificant at the 1% significance level (99% confidence level) with t-value = 0.0051. The average

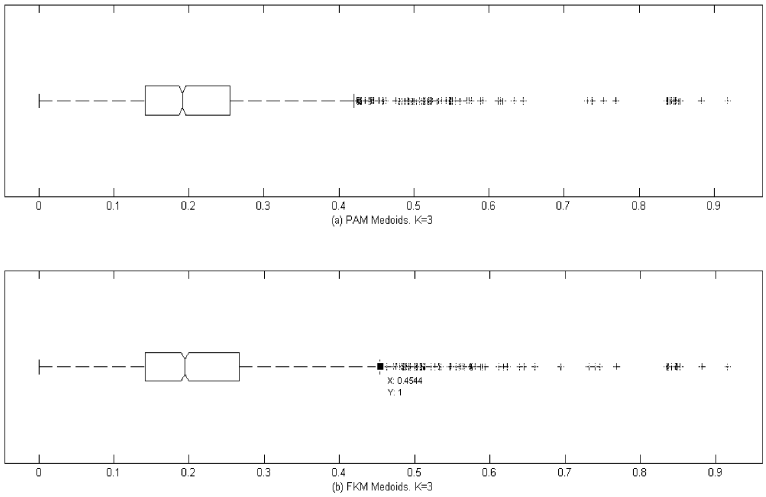


Fig. 5: Box plot of differences between data objects of the Yeast dataset and the medoids produced using the PAM and the FKM methods when K=3.

difference between the distance achieved using PAM and FKM is 0.09 which indicates that similar quality results are achieved using the two methods. The main advantage of the FKM method is the fast execution time as shown in Table 6.

For the Wood and Bupa datasets, both methods achieved the same performance as the same mediods are determined using the two methods. The FKM is much more efficient in terms of CPU time as illustrated in Table 6.

For the Yeast dataset, very close results were achieved with difference in AD = 0.0042 with t-value equals -0.0029 which indicates the differences are statistically insignificant.

Figures 5 (a) and (b) show the box plots of differences between the data points of the Yeast dataset and the corresponding medoids produced using the PAM method and the proposed FKM method, respectively [24].

The results achieved by PAM and FKM methods are almost the same as shown in Figure 5 (a) and (b), respectively. From Figure 5 (a) it can be noticed that when PAM method is used with k=3, 25% of the differences for the first quartile of the data points are below 0.1421. The median difference has the value of 0.1913. The maximum difference for the third quartile is 0.2546. The maximum difference for the majority of points is 0.4241 with few outliers of higher differences.

With the same number of cluster k=3, when FKM method is used as shown in Figure 5 (b), the differences of the first quartile of data points are below 0.1415. The median difference has the value of 0.1947.

Table 6: CPU run time in milliseconds for different high dimensional datasets

Data set	PAM	FKM	Improvement
Iris	172	94	45%
Wood	94	16	83%
Bupa	562	78	86%
Yeast	33390	1250	96%

Table 7: The AD values produced by the PAM and the FKM methods when applied on the Bupa dataset

K	PAM	FKM	Difference	t-value
2	29.2056	29.2056	0	NA
4	23.6969	24.1014	0.4045	-0.0012
6	20.6599	20.9554	0.2955	0.0004
8	18.9117	19.2660	0.3543	-0.0048
10	17.5809	18.2879	0.7070	0.0058

The maximum difference for the third quartile is 0.2666. The maximum difference is 0.4544 with more outliers than in Figure 5 (b) which justifies the differences between the two methods. Also, the range of the fourth quartile is wider than the range of the fourth quartile of the PAM method which indicates that few data points affect the average distance; nevertheless, the difference in the results obtained from the two methods is statistically insignificant. More importantly, the results obtained from the FKM method is more efficient than the results obtained from the PAM method in terms of the CPU time.

Table 6 shows the results in terms of the CPU running time for both the FKM and PAM methods. The table shows that in all cases, the results of the proposed

method were better than those of the PAM method in terms of the CPU time. This efficiency increases monotonically with the increase in the size of the dataset; it reaches the maximum improvement of 96% for the Yeast dataset which has 1484 data objects.

Finally, we will test our proposed method against PAM when a number of clusters are applied on high dimensional datasets. We have chosen the Bupa dataset since its size is moderate. Table 7 shows the result.

As Table 7 shows, the results obtained from the FKM method is similar to the results obtained from the PAM method as in all cases the differences are statistically insignificant at the 99% confidence level. The main advantage is that the proposed FKM method is much faster than the PAM method.

CONCLUSIONS

Partitioning clustering is an important part of cluster analysis. In this paper, we have proposed an efficient fuzzy partitioning clustering method called Fuzzy k-Medoids method (FKM), based on the FCM algorithm. The medoids are selected as the objects with the highest membership grade in each cluster.

To test the performance of the proposed method, different benchmarked datasets obtained from the University of California, Irvine repository of machine learning databases have been used with different number of clusters and different dimensionality.

In terms of the quality of the results, the proposed FKM method performs similarly to the benchmarked PAM method with no statistical significance between the performances of the two methods.

In terms of the CPU running time, the proposed FKM method outperforms the PAM method in all cases, this CPU efficiency is more apparent when datasets of high dimensional or datasets with large number of data objects are applied. This makes the proposed method a good choice for solving the k-medoids problem, particularly when high dimensional data sets are involved in applications such as Geographic Information Systems (GIS) and gene expression.

Enhancing the quality of the results is left to future work.

REFERENCES

1. Jain, A., M. Murty and P. Flynn, 1999. Data clustering: a review. ACM Computing Surveys (CSUR), 31(3): 264-323.
2. Ng, R. and J. Han, 2002. CLARANS: A Method for Clustering Objects for Spatial Data Mining. IEEE Transactions Knowledge and Data Eng., 14(5): 1003-1016.
3. Jain, A. and R. Dubes, 1988. Algorithms for Clustering Data, Prentice-Hall, Upper Saddle River, USA.,
4. MacQueen, J., 1967. Some Methods for Classification and Analysis of Multivariate Observations. In the Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics, pp: 281-297.
5. Zadeh, L., 1965. Fuzzy Sets, Information and Control, 8: 338-353.
6. Sheng, W. and X. Liu, 2006. A Genetic K-Medoids Clustering Algorithm, J. Heuristics, 12: 447-466.
7. Xu, R. and D. Wunsch II, 2005. Survey of Clustering Algorithms, IEEE Transactions on Neural Networks, 16(3): 645-678.
8. Garg, S. and R. Jain, 2006. Variations of K-mean Algorithm: A Study for High-Dimensional Large Data Sets. Information Technology J., 5(6): 1132-1135.
9. Kaufman, L. and P.J. Rousseeuw, 1987. Clustering by Means of Medoids. In Statistical Data Analysis Based on the L_1-Norm and Related Methods, edited by Y. Dodge, North-Holland, Amsterdam, pp: 405-416.
10. Kaufman, L. and P.J. Rousseeuw, 1990. Finding Groups in Data: An Introduction to Cluster Analysis. J. Wiley and Sons, New York.
11. Ng, R. and J. Han, 1994. Efficient and Effective Clustering Methods for Spatial Data Mining. In the Proceedings of the 20th Very Large Databases Conference (VLDB 94), Santiago, Chile, pp: 144-155.
12. Campello, R., E. Hruschka and V. Alves, 2009. On the efficiency of evolutionary fuzzy clustering, J. Heuristics, 15: 43-75.
13. Han, J. and M. Kamber, 2006. Data Mining: Concepts and Techniques, 2nd edition. Morgan Kaufmann, San Fransisco, USA.,
14. Kolatch, E., 2001. Clustering Algorithms for Spatial Databases: A Survey, Dept. of Computer Science, University of Maryland, College Park. Available: http://citeseerx.ist.psu.edu/436843.html. (Visited on July 2010).
15. Dave, R. and R. Krishnapuram, 1997. Robust Clustering Methods: A Unified View. IEEE Transactions on Fuzzy Systems, 5(2): 270-293.
16. Motameni, H. Movaghar, A. Daneshfar, I. Zadeh, H. and J. Bakhshi, 2008, World Applied Sciences J., 3(3): 514-521.

17. Razavi, S., T. Mehran and E. Kaber, 2008, Improving in performance Neural Network for Persian Handwritten Digits Recognition using FCM Clustering, World Applied Sciences J., 5(2): 150-160.
18. Yu, J., 2005. General C-Means Clustering Model. IEEE Transactions on Pattern Analysis and Machine Intelligence, 27(8): 1197-1211.
19. Cherkassky, V. and F. Mulier, 1998. Learning from Data: Concepts, Theory and Methods, John Wily and Suns, New York, USA.,
20. Bezdek, J., 1981. Pattern Recognition with Fuzzy Objective Function Algorithms, Plenum press, New York, USA.
21. Bezdek, J., R. Ehrlich and W. Full, 1984. FCM: The Fuzzy C-Means Clustering Algorithm, Computers and Geosciences, 10(2-3): 191-203.
22. Chiang, J. and Z. Yin, 2007. Unsupervised Minor Prototype Detection Using an Adaptive Population Partitioning Algorithm, Pattern Recognition, 40: 3132 - 3145.
23. Anderson, D.R., D.J. Sweeney and T.A. Williams, 2010. Statistics for Business and Economics, 11th edition, South-Western.
24. Ilaboya, I.R., F.F. Asekhame, M.O. Ezugwu, A.A. Erameh and F.E. Omofuma, 2010. Studies on Biogas Generation from Agricultural Waste; Analysis of the Effects of Alkaline on Gas Generation, World Applied Sciences J., 9(5): 537-545.
25. Frank, A. and A. Asuncion, 2010. UCI Machine Learning Repository, Irvine, CA: University of California, School of Information and Computer Science. Available: http://archive.ics.uci.edu/ml (Visited on August 2010).