World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:3, No:1, 2009

Fuzzy Relatives of the CLARANS Algorithm With Application to Text Clustering

Mohamed A. Mahfouz, and M. A. Ismail

Open Science Index, Computer and Information Engineering Vol:3, No:1, 2009 publications.waset.org/410/pdf

***Abstract*—**This paper introduces new algorithms (Fuzzy relative of the CLARANS algorithm FCLARANS and Fuzzy c Medoids based on randomized search FCMRANS) for fuzzy clustering of relational data. Unlike existing fuzzy c-medoids algorithm (FCMdd) in which the within cluster dissimilarity of each cluster is minimized in each iteration by recomputing new medoids given current memberships, FCLARANS minimizes the same objective function minimized by FCMdd by changing current medoids in such away that that the sum of the within cluster dissimilarities is minimized. Computing new medoids may be effected by noise because outliers may join the computation of medoids while the choice of medoids in FCLARANS is dictated by the location of a predominant fraction of points inside a cluster and, therefore, it is less sensitive to the presence of outliers. In FCMRANS the step of computing new medoids in FCMdd is modified to be based on randomized search. Furthermore, a new initialization procedure is developed that add randomness to the initialization procedure used with FCMdd. Both FCLARANS and FCMRANS are compared with the robust and linearized version of fuzzy c-medoids (RFCMdd). Experimental results with different samples of the Reuter-21578, Newsgroups (20NG) and generated datasets with noise show that FCLARANS is more robust than both RFCMdd and FCMRANS. Finally, both FCMRANS and FCLARANS are more efficient and their outputs are almost the same as that of RFCMdd in terms of classification rate.

***Keywords*—**Data Mining, Fuzzy Clustering, Relational Clustering, Medoid-Based Clustering, Cluster Analysis, Unsupervised Learning.

I. Introduction

RELATIONAL data refers to the situation where we have only numerical values representing the degrees of similarity(or dissimilarity) between each pair of objects in the data. While object data refers to the situation where the objects to be clustered are represented by vectors $x_i \in \Re p$. Algorithms that generate partitions of relational data are usually referred to as relational clustering algorithms.

Medoid-based algorithms like PAM and CLARA proposed by Kaufman and Rousseeuw [3], [4] respectively are examples of relational clustering algorithms in which a cluster is represented by one of its objects instead of computing centers for each cluster from its objects as centeroid-based algorithms which makes medoid-based algorithms applicable to relational data, further more these medoids have embedded resistance against outliers since peripheral cluster points do not affect them. When medoids are selected, clusters are defined as subsets of points close to respective medoids, and the objective function is defined as the averaged distance or another dissimilarity measure between a point and its medoid.

CLARA uses several (five) samples, each with 40+2k points, which are each subjected to PAM. The whole dataset is assigned to resulting medoids, the objective function is computed, and the best set of medoids is retained.

Further progress is associated with Ng and Han [5] who introduced the algorithm CLARANS (Clustering Large Applications based upon RANdomized Search) in the context of clustering in spatial databases. Authors considered a graph whose nodes are the sets of k medoids and an edge connects two nodes if they differ by exactly one medoid. While CLARA compares very few neighbors corresponding to a fixed small sample, CLARANS uses random search to generate neighbors by starting with an arbitrary node and randomly checking maxneighbor neighbors. If a neighbor represents a better partition, the process continues with this new node. Otherwise a local minimum is found, and the algorithm restarts until numlocal local minima are found.

Other algorithms for relational clustering include Gowda and Diday [7], Ramkumar and Swami [8] ,El Sonbaty and Ismail [9] , and Bajcsy and N. Ahuja [10].

More recent algorithm for relational clustering in which Zhang and Couloigner [17] presented a k-medoids algorithm for spatial clustering in large applications (CLATIN) that utilizes the TIN of medoids to facilitate local computation when searching for the optimal medoids.

PAM, CLARA, CLATIN, CLARANS and Hierarchical algorithms as in [1], [2] generate crisp clusters. When the clusters overlap as the case in text clustering, we may desire fuzzy clusters.

One of the early fuzzy relational clustering algorithms are the ones due to Ruspini [11]. Hathaway *et al.* [12] introduced Relational Fuzzy C-Means (RFCM). The NERFCM model [25] extends RFCM to ease the restrictions that RFCM imposes on the dissimilarity matrix. More recently, Dav´e *et al.* [19] generalize this approach further by including an extension to handle data sets containing noise and outliers.

Manuscript received September 15, 2008.

M. A. Mahfouz is a PhD candidate in the Department of Computer Science and Systems Engineering, Faculty of Engineering, Alexandria, 21544 Egypt (phone:+2012-481-2622; e-mail: mohamed.mahfouz @eng.alex.edu.eg).

M.A. Ismail is Professor of Computer Science and Former Dean, Faculty of Enginneering, University of Alexandria, Alexandria 21455, Egypt.(He is now President of Pharos University in Alexandria) (e-mail: maismail@pua.edu.eg).

Open Science Index, Computer and Information Engineering Vol:3, No:1, 2009 publications.waset.org/410/pdf

The study most relevant to our focus here is [22]. Krishnapuram *et al.* [22] proposed a fuzzy clustering for a relational data termed as FCMdd (Fuzzy C-Medoids) and its robust version (RFCMdd). In [22] RFCMdd is compared with the Relational Fuzzy C-Means algorithm (RFCM) and showed that RFCMdd is more efficient. Recently Maji and Sankar[13] applied the principles of rough sets, fuzzy sets[15] to the c-medoids algorithm[22] and proposed rough-fuzzy c-medoids algorithm, to select the most informative bio-bases[14] and the amino acid mutation matrix[16] is used in computing the similarity between objects(sequences).

However, the linearized version of FCMdd is O (n log n) this complexity still unsuitable for text clustering where datasets are extremely large. To deal with this problem, this paper presents a new algorithm FCLARANS relative of the famous medoid-based algorithm CLARANS for fuzzy clustering of relational data. Unlike existing fuzzy c-Medoids algorithm (RFCMdd) in which the averaged distance between an object and its medoid (the within cluster dissimilarity) is minimized in each iteration by recomputing new medoids given current memberships instead the objective function is minimized by changing current medoids such that the within cluster dissimilarity is minimized. Experimental results show that FCLARANS compares favorably with RFCMdd in terms of classification rate (Fmeasure) but much faster than RFCMdd.

Also another variation of RFCMdd is proposed in which the step of computing new medoids in RFCMdd is modified to be based on randomized search termed as FCMRANS which output almost the same quality as RFCMdd but at the same time is an order of magnitude faster.

The rest of the paper is organized as follows: section II; reviews fuzzy medoid-based clustering and describes the algorithm RFCMdd [22]. Section III; describes the proposed algorithms and the proposed initialization procedure. Section IV; describes the experiments done for tuning the input parameters used in FCMRANS, and FCLARANS. Section V; compares the performance of the proposed algorithms with RFCMdd[22]. Finally section VI concludes the paper with summary and ideas for future work.

II. Related work

A. Fuzzy Medoid-Based Clustering

Let X = $\{x_1, x_2, x_3 \ldots x_n\}$ be a set of n patterns each pattern may or may not be represented by a feature vector. Let $d(x_i,x_j)$ denotes the dissimilarity between patterns x_i and x_j. Let V = $\{v_1, v_2, \ldots\ldots, v_k\}$, $v_i \in X$ represents a subset of X with cardinality k , i.e. V is a subset of X. Let X^k represents the set of all c-subsets V of X.

The objective function minimized medoid-based algorithms is the average distance between an object and medoids of the clusters in which it has a membership greater than zero defined as follow:-

$$\text{Minimize: } J_m(V;X)=\sum_{j=1}^{n}\sum_{i=1}^{k} U_{ij}^{m}\, d(x_j,v_i) \quad (1)$$

Where the minimization is performed over all V in X^k. The membership u_{ij} is the fuzzy membership of x_j in cluster i and can be defined heuristically in several ways. We will use the FCM [18] membership model given by:

$$u_{ij} = \frac{\left(1 / d(x_j, v_i)\right)^{1/m-1}}{\sum_{c=1}^{k}\left(1 / d(x_j, v_c)\right)^{1/(m-1)}} \quad (2)$$

Where m $\in$ [1, ∞) is the fuzzifier. Other possibilities for defining memberships can be found in [22].

B. Robust Fuzzy C-Medoids Algorithm (RFCMdd) [22]

RFCMdd starts by selecting initial medoids and in each iteration it computes memberships for all objects in all clusters then from computed membership it tries to compute a new medoid for each cluster based on the memberships of the objects in that cluster, if computed medoids is the same as previously computed medoids it terminates.

It is clear that outliers freely join the computations of medoids. To overcome this problem, the objective function in [22] is modified to decrease the chance that outliers join the computation as follow:

$$J_m(V;X) = \sum_{x_j \in h}\Big(\sum_{i=1}^{k}\big(d(x_j,v_i)\big)^{1/1-m}\Big)^{1-m} \quad (3)$$

Where h is a subset of the original dataset after excluding objects x_j having harm(x_j) greater than a threshold.

$$\text{harm}(x_j) = \Big(\sum_{i=1}^{k}\big(d(x_j,v_i)\big)^{1/1-m}\Big)^{1-m} \quad (4)$$

This allows the clustering algorithm to ignore outlier objects while computing the objective function.

The threshold is chosen depending on the expected percentage of outliers in the original dataset for example if we expect 30% outliers we choose the threshold such that 30% of the data are excluded from the computation.

The robustness of the algorithm is dependent on selecting a proper value for the threshold. A prior knowledge about the percentage of outliers in the dataset is needed.

Table I describes the RFCMdd presented in [22]. They reduced the complexity of the algorithm by considering only the objects near current medoids as candidate medoids in the computation of medoids and increase the robustness by considering only objects in the set *h* in the computation of the objective function.

Open Science Index, Computer and Information Engineering Vol:3, No:1, 2009 publications.waset.org/410/pdf

TABLE I
ROBUST FUZZY C-MEDOIDS ALGORITHM (RFCMDD)

Begin [RFCMdd]
1. *iter*=0
2. Set a value for fuzzifier *m*
3. Initialize the set of current medoids V={ $v_1, v_2, ., v_k$ }
4. **Repeat**
4.1.Compute memberships u_{ij} for all *i,j*
4.2 Compute the set *h* using (4) /*exclude outliers*/
4.3.V^{old} = V /*Store medoids*/
4.4.**For** *i* = 1 to *k* do /*Compute new medoids */
4.4.1. compute *Xp*(*i*) the set of *p* objects having the highest membership in cluster i./*neighbors*/
4.4.2. $q = \underset{x_k \in X_{(p)i}}{\operatorname{argmin}} \sum_{x_j \in h} u_{ij}^m \, d(x_k, x_j)$
4.4.3. $v_i = x_q$
Endfor
4.5 *iter* = *iter* +1
Until (*iter* > *maxiter* or V^{old} = V)
End [RFCMdd]

TABLE II
FUZZY C-MEDOIDS BASED ON RANDOMIZED SEARCH (FCMRANS)

Begin [FCMRANS]
1. *iter*=0, *maxneighbor*= 1.25%k(n-k)
2. set values of fuzzifier *m* and maxneighbor
3. Initialize the set of current medoids V={ $v_1, v_2, ., v_k$ }
4. **Repeat**
4.1.Compute memberships u_{ij} for all *i,j*
4.2 Compute the set *h* using (4) /*exclude outliers*/
4.3.V^{old} = V /*Store medoids*/
4.4. **Repeat**
4.4.1. Change one of the current medoids v_c in V with not medoid object x_m, give a higher probability of selection to objects $x_m \in Xp(i)$.
4.4.2. *j* = *j* +1
4.4.3. Calculate the cost of the change *E* using(5).
4.4.4. If (*E*<0) Then
Remove the changed medoid v_c from V and add x_m to V .
j = 1.
Endif
Until (*j* > maxneighbor)
4.5 *iter* = *iter* +1
Until (*iter* > *maxiter* or V^{old} = V)
Until (*iter* > *maxiter* or V^{old} = V)
End[FCMRANS]

The original version of the algorithm FCMdd is the same except that all objects are included in the computation of the objective function and are considered candidate medoids in the computation of medoids.

III. THE PROPOSED ALGORITHMS

A. Fuzzy C-Medoids based on Randomized Search(FCMRANS)

Table II describes FCMRANS. It is the same as RFCMdd except the step 4.4 of computing new medoids is modified to be based on randomized search to reduce the runtime.

Current medoids are replaced by non medoids objects one at a time such that the change should reduce the within cluster dissimilarity of the cluster of changed medoid based on previously computed memberships. The cost of replacing a current medoid v_k by non medoid object x_m is defined as follow:

$$E = \sum_{i=1}^{s} (d(x_i, x_m) - d(x_i, v_k)) u_{ki}^m \quad (5)$$

To add randomness to the step of computing new medoids instead of only considering neighborhood of current medoids as in [22] , candidate medoids are selected randomly such that the neighborhood of a current medoid is given higher probability for selection inversely proportional to their distance from current medoid. The added randomness reduces the possibility that the algorithm stuck in local minima.

B. Fuzzy Relative of CLARANS (FCLARANS)

This section describes the proposed algorithm FCLARANS that starts by selecting initial medoids same as RFCMdd and then try to change one of the current medoids by randomly selecting one of the not-medoid objects same as CLARANS. FCLARANS don't compute the medoid instead we do backtracking trying to change one of our old choices by another that should minimize the objective function, since FCLARANS don't compute new medoids the algorithm is not effected by outliers since outliers will never represents better choices.

The cost function is different from the cost function of FCMRANS. The cost of a change E in FCLARANS is the difference in the objective function defined as the sum of the average distances within all clusters before and after a change thus temporarily memberships are computed.

$$E = \sum_{j}^{n} \sum_{i}^{k} d(x_j, x_m) \mu_{ij}^{m'} - d(i, v_i) \mu_{ij}^m \quad (6)$$

Where $\mu_{ij}^{m'}$ is the new membership of object x_j in cluster *i* given that x_m is the new medoid of cluster *i*. The temporary memberships can be calculated efficiently by computing and storing $\left(1/d(x_j, v_i)\right)^{1/m-1}$ for all objects instead of storing the memberships. Each time we change a medoid v_i with not

Open Science Index, Computer and Information Engineering Vol:3, No:1, 2009 publications.waset.org/410/pdf

medoid object x_m we compute and store $\left(1/d(x_j, x_m)\right)^{1/m-1}$ for all x_j.

TABLE III
FUZZY RELATIVE OF CLARANS ALGORITHM (FCLARANS)

Begin [FCLARANS]
1. *iter*=0, *maxneighbor*= 1.25%k(n-k)
2 set values of fuzzifier *m* and maxneighbor
3. Initialize the set of current medoids V={ v_1,v_2,., v_k }
4. Compute memberships u_{ij} for all *i, j* by using (2).
5. **Repeat**
5.1. Change one of the current medoids *vc* in V by one of the not medoid objects x_m give higher probability of selection to neighbors of v_c.
5.2. Calculate a temporary memberships for all objects given the above change in medoids.
5.3. $j = j + 1$
5.4. Calculate the cost of the change *E* using (6).
5.5. **If** (*E*<0)
Remove the changed medoid v_c from V
Set the temporary memberships as current.
Add x_m to V .
$j = 1$
Endif
Until (*j* > *maxneighbor*)
End [FCLARANS]

C. Effect of Initial Medoids

FCLARANS similar to its hard version CLARANS[5] is less sensitive to initial condition than RFCMdd and FCMRANS. Krishnapuram *et al.* [22] experimented with three strategies for selecting initial medoids:

1) Selecting all initial k medoids randomly .
2) Selecting the first initial medoid as the object that is the most central to the dataset and then selecting the next initial medoid as the object that is most dissimilar to previously selected medoids until k medoids selected.
3) Selecting the first initial medoid randomly and then selecting the next initial medoid as the object that is most dissimilar to previously selected medoids until k medoids selected.

They found that both second and third initialization procedure work well.

This section proposes a new initialization procedure (InitRand) by adding randomness to their third initialization procedure as follows:

Select the first initial medoid randomly and select the next initial medoid randomly from the objects that are neither the most closest nor the most dissimilar to previously selected medoids until k medoids selected.

Table IV shows the description of the algorithm. In step 3.3, random() is a function that returns a random value in [0,1]. The third initialization procedure in [22] is the same as ours except that in step 3.3 the object that has a higher distance than the currently chosen medoid will always replace it.

TABLE IV
INITIALIZATION PROCEDURE FOR SELECTING INITIAL MEDOIDS (INITRAND)

Begin [InitRand]
1. Select First Medoid v_1 randomly;
2. V = {v_1}, *iter* = 1;
3. **Repeat**
3.1 *iter* = *iter*+1;
3.2 *maxdist* = 0
3.3 **For** *i* = 1 to n
$\text{dist} = \min_{1 \le c \le |v|} d(v_c, x_i)$;
If (*dist* > *maxdist*)
If (**random**()<((*dist* – *maxdist*)/*dist*)
q=*i*;
maxdist=*dist;*
Endif
Endif
Endfor
3.4 $v_{iter} = x_q$
3.5 $V = V \cup \{v_{iter}\}$
Until *iter*=*k*
End[InitRand]

The proposed initialization procedure decreases the chance that the algorithm get stuck in local minimum. Experiments with RFCMdd, FCMRANS and FCLARANS on different datasets using the new initialization procedure (InitRand) show that the average runtime is lower for the three algorithms than using the third initialization procedure also the quality is improved when we try to find more than one local minima.

Ester[23] proposed three focusing techniques employing R*-trees in order to make CLARANS more efficient for large databases. Recently Camila *et al.* [24] presented a new algorithm, PAM-SLIM, which employs metric access methods to scale-up medoid-based algorithms. All these techniques can be applied to FCLARANS, in a preprocessing step, after the tree is built objects that will be considered as candidate medoids are selected, and only those objects will be considered in the step of selecting a new object to replace a current medoid.

IV. EXPERIMENTAL RESULTS: TUNING

Both fuzzy FCLARANS and FCMRANS are based on randomized search[5] so we need to find appropriate value for percentage of neighbors maxneighbor. Also both FCLARANS, FCMRANS and RFCMdd are fuzzy algorithms so we need to find appropriate value for the fuzzifier m and finally they are not guaranteed to find the global minimum so we need to find more than local minimum starting from different initial state and select best clustering among local

Open Science Index, Computer and Information Engineering Vol:3, No:1, 2009 publications.waset.org/410/pdf

minimums reached, the parameter numlocal represents the number of local minimums we need to compute to reach high quality solution with minimum runtime.

We experimented with generated datasets whose clusters are known. Each dataset consists of k clusters where k=2,4,6,8,10 and n objects where n=1000,2000,..5000. The symbol n-k used to represents a generated dataset with n objects in k clusters.

The best value for a parameter is the one that often give the lowest average runtime and the highest average quality among the generated datasets. Every time a suitable value for a parameter is computed it is used in later experiments.

To measure the quality of the clustering produced for generated dataset we use validity index V_{XB} proposed by Xie and Beni [21]. V_{XB} focuses on two properties: compactness and separation The numerator part of Eq. (7) indicates the compactness of fuzzy partition. The denominator part indicates the strength of separation between clusters.

$$V_{XB} = \sum_{j=1}^{n}\sum_{i=1}^{k} u_{ij}^{m} d(x_j, v_i) / (n \min_{i \neq k} d(v_i, v_k)) \quad (7)$$

A. Determining the Maximum Number of Neighbors

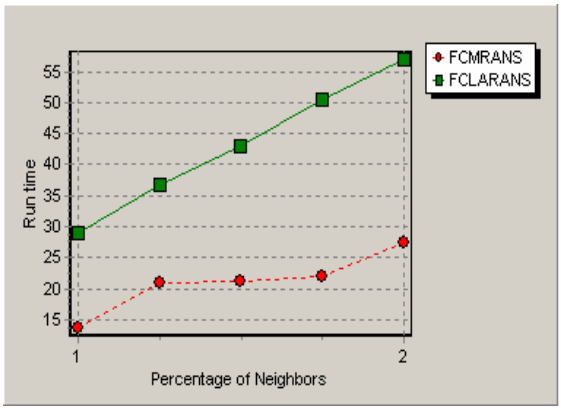


Fig. 1 Runtime for Different values for maxneighbor

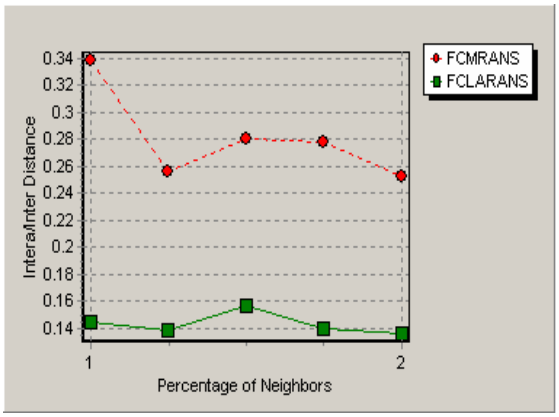


Fig. 2 Intera/Inter distance for different values for maxneighbor

Fig. 1-2 show that changing the maxneighbor over 1.25%k(n-k) increases the runtime while no significant change in the quality of the quality measured as the intra/inter cluster distance(V_{XB}). This value will be used in the next experiments.

B. Determining the Number of Local Minima

Fig. 3-4 shows that finding more than two local minima increases the runtime too much while no significant change in the quality.

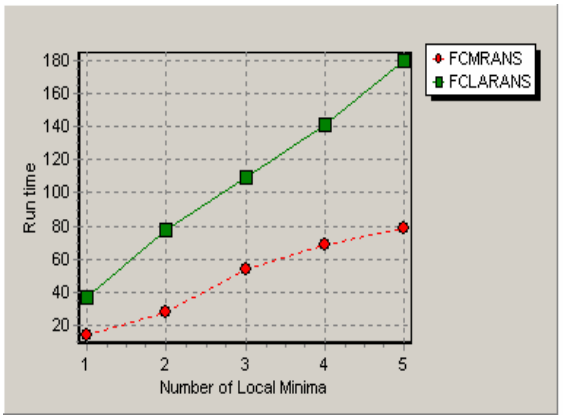


Fig. 3 R untime for different values for numlocal

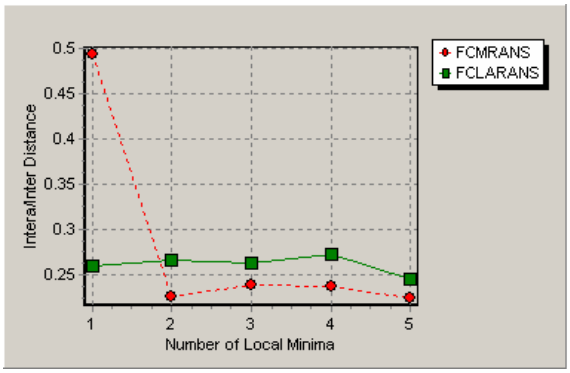


Fig. 4 intera/inter cluster distance for different values for numlocal

C. Determining the Value of the Fuzzifier m

Experimentation with different value for m=1.1, 1.3, 1.5, 1.7, 2,2.5,3 on datasets rn-k are conducted for small, medium and large values of n and k. A value of 1.5 is found to be appropriate for the fuzzifier *m*.

The fuzzifier m in the above algorithms determines the degree of fuzziness of the resulting clusters. In [22], [13] they recommend the same value for the fuzzifier and show that the reason is that the medoid always has a membership of 1 in the cluster, raising its membership to the power m has no effect. Thus when m is high (m>>1) the mobility of the medoids from iteration to iteration may be lost, because all memberships become very small except the one corresponds to the current medoid.

Experimentation with RFCMdd, FCLARANS, and FCMRANS with different values of m , on different datasets show that as m increases the partition coefficient[21] decreases and the partition entropy [21] increases and the runtime slightly changes but the quality as intera/inter-cluster distances show that a value of 1.5 is appropriate for m and at

Open Science Index, Computer and Information Engineering Vol:3, No:1, 2009 publications.waset.org/410/pdf

the same time gives suitable values for both the partition entropy and the partition coefficient.

V. Performance Evaluation

A. Datasets and Preprocessing

To reduce the risk that our conclusions might be valid only on a particular corpus, we used two distinct test corpora: a subset of the 20 Newsgroups (20Ng) data set [23], a subset of the Reuters-21578 [20] and generated datasets with embedded outliers to test the robustness of the algorithms.

The 20 Newsgroups (20Ng) data set, collected by Ken Lang, contains about 20,000 articles evenly divided among 20 UseNet Discussion groups. Each newsgroup represents a category.

In Reuter many stories belong to more than single category, only single-category documents are considered which make up 66 different categories of different sizes. The 3 biggest categories contain more than 70% of all single-topic documents. The 10 biggest categories contain more than 86% of all single-topic documents. To Account for these biased a-priory probabilities, the investigated test sets were constructed as a uniform distribution with c different topics, where c is the number of topics to experiment with.

To render the representations of the documents more precisely, stop words such as "I", "am", "is" etc. are removed. Remaining terms were reduced by means of porter's stemming algorithm by more than 30% for example "engineer" "engineering" reduced to "engine".

As Most text clustering algorithms we rely on so- called vector space model. In this model each text document d is represented as a vector of frequencies of the remaining m terms:

$$d = (tf_1, tf_2, tf_3, ..., tf_m)$$

In our experiments the document vectors are normalized to unit length to allow comparison of documents of different lengths.

In each run, 20% of the documents were randomly selected from the collection, and only these documents were used to generate the keywords and the 10 eigenvectors. The 500-dimensional feature vectors were constructed for all the remaining documents, and these feature vectors were then projected onto the 10 eigenvectors to generate the object data for a particular run of the algorithms.

B. Quality Measures

When correct clusters are known as the case with labeled datasets this allowed us to use the F-Measure method to calculate the classification rate as an additional measure of quality. F-Measure combines the precision and recall ideas from information retrieval. The following is a procedure for calculating F-Measure for the output of fuzzy algorithms.

Let $C = \{C_1, C_2, C_3, \ldots, C_k\}$ be the set of output clusters of the clustering algorithm for dataset D.

Let $C^* = \{C_1^*, C_2^*, ..., C_l^*\}$ be the set of correct clusters for dataset D.

Then the recall of cluster j with class *i* ,*rec(i,j),* is :

$$\text{rec}(i, j) = \left|C_j \cap C_i^*\right| / \left|C_i^*\right| \quad (8)$$

The precision of cluster j with respect to class I, *prec*(*i, j*) is :

$$\text{prec}(i, j) = \left|C_j \cap C_i^*\right| / \left|C_J\right| \quad (9)$$

$$F_{i,j} = 2.pred(i, j).rec(i, j)/(pred(i, j) + rec(i, j)) \quad (10)$$

$$F = 1/|D| \sum_{i=1}^{l} \left|C_i^*\right| . \max_{j=1,\ldots,k} \{F(i, j)\} \quad (11)$$

In fuzzy clustering we can defuzzify the output by assigning each object to the cluster with which the object has the highest membership and use the measure in (11). We choose to modify (8), (9) to take the memberships into account as follow:

$$\text{rec}(i, j) = \sum_{x_k \in C_i^*} u_{kj} / \left|C_i^*\right| \quad (12)$$

$$\text{prec}(i, j) = \sum_{x_k \in C_i^*} u_{kj} / \sum_{i=1}^{n} u_{ij} \quad (13)$$

Where *n* is the size of the dataset.

C. Effect of Changing the Noise Level

We ran the three algorithms with four noise levels range from 0.05 to 0.020 embedded into generated dataset of size 3000 and 10 attributes, represents 10 clusters.

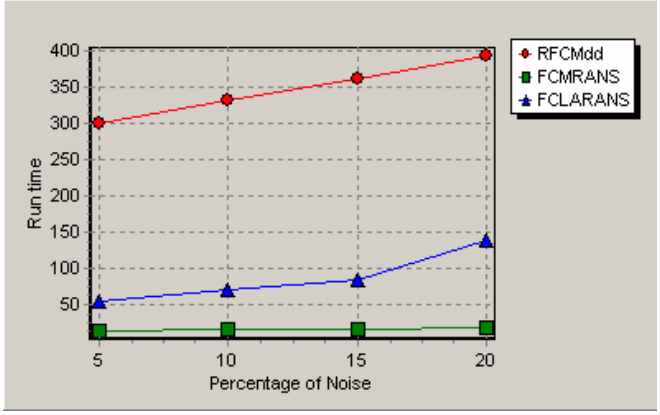


Fig. 5 Runtime for different noise level

Fig. 5 shows that that runtime for FCLARANS is much less than RFCMdd while it gives higher quality in noise level ranges from 0.05 to 0.2 in terms of intera/inter cluster distance and FCMRANS is faster than both FCLARANS and RFCMdd. The quality of FCMRANS is less than that obtained for FCLARANS and similar to that obtained for RFCMdd but it gives the worst quality for the %20 percentage of noise.

Open Science Index, Computer and Information Engineering Vol:3, No:1, 2009 publications.waset.org/410/pdf

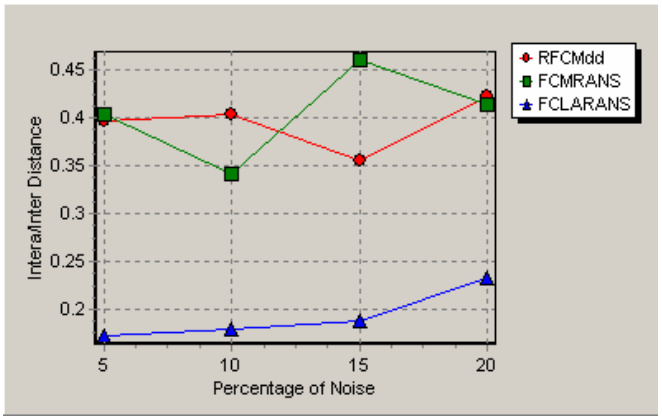


Fig. 6 Intera/Inter Cluster distance for different noise levels

D. *Changing the Dataset Size*

In this experiment, a set of 1000, 2000, 3000 and 4000 are uniformly selected from the top ten categories in Reuter dataset.

Fig. 7 shows that Both FCLARANS and FCMRANS were more efficient than RFCMdd for all dataset sizes. Both FCMRANS and FCLARANS are always less than RFCMdd and the difference increases as the dataset size increases. FCMRANS was the most efficient.

Fig. 8 shows the quality of the produced clusters measured as the classification rate F-measure. The three algorithms have almost the same classification rate.

Other quality measures such as intra/inter cluster distance, partition coefficient and partition entropy [21], are measured. However, only classification rate is presented due to space constraints. The three algorithms have approximately the same values for these measures.

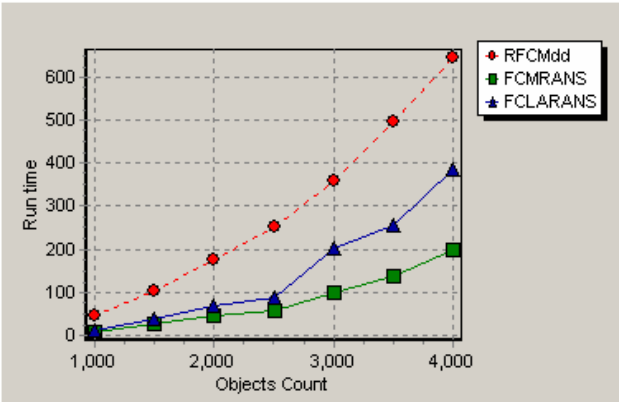


Fig. 7 Runtime for different dataset sizes of Reuter from ten categories

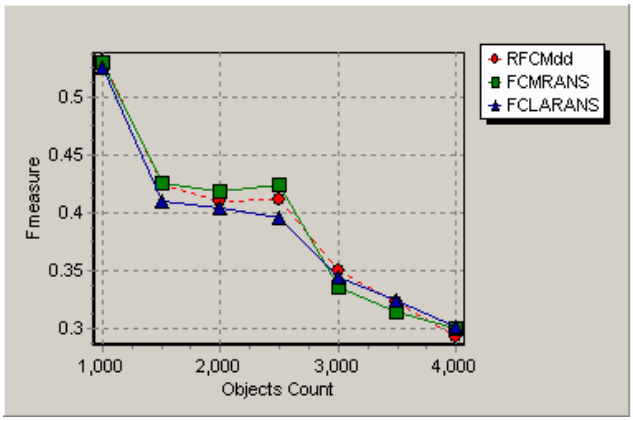


Fig. 8 Fmeasure for different dataset sizes of Reuter from ten categories

E. *Changing the Number of Clusters*

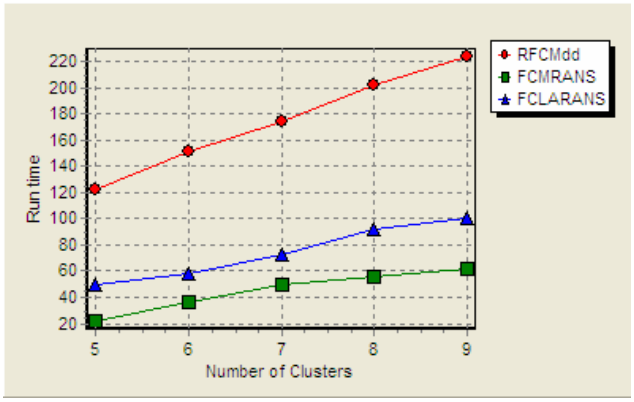


Fig. 9 Runtime for different cluster count on sample from NG of size 2500

In this experiment, 2500 documents from News Groups are uniformly selected from 2-9 categories.

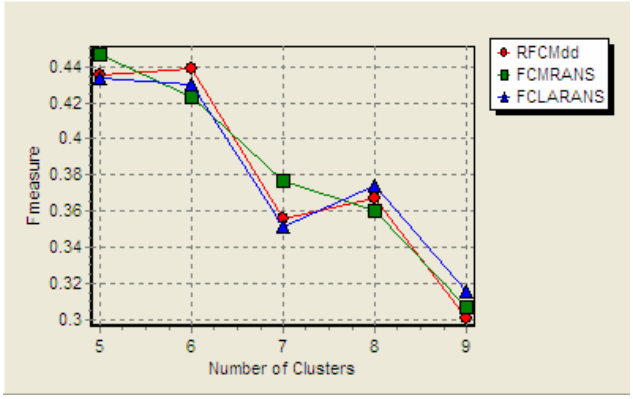


Fig. 10 F-measure for different cluster count on from NG of size 2500

Fig. 9 shows the runtime for different cluster count. The runtime of the three algorithms increase as the number of clusters increases as the complexity analysis shows. The runtime of both FCLARANS and FCMRANS is always less than RFCMdd. Fig. 10 shows that the proposed algorithms

Open Science Index, Computer and Information Engineering Vol:3, No:1, 2009 publications.waset.org/410/pdf

give almost the same Fmeasure as that of RFCMdd for different number of clusters.

VI. Conclusions

This paper presented new relational fuzzy clustering algorithms based on randomized search. The complexities of the proposed techniques compare favorably with that of the RFCMdd algorithm. Our preliminary results show that both FCMRANS and FCLARANS algorithms are more efficient than RFCMdd when applied to large datasets. The following can be concluded from the results:

1) Both FCMRANS and FCLARANS produce almost the same quality of results as RFCMdd. However, both techniques are an order of magnitude faster than RFCMdd. This means that the difference increases as the data size increases.
2) Introducing initialization procedure for medoids, enhances the quality of the results of the above algorithms.
3) Increasing in the fuzzifier value does not increase the runtime rapidly as in fuzzy c-means algorithm, and the appropriate value was found to be less than 2.
4) FCLARANS is found to be less sensitive to outliers than RFCMdd and FCMRANS
5) Appropriate values for maxneighbor and numlocal parameters of the proposed algorithms are found to be similar to the values in [5] which are 1.25 and 2 respectively.

Other applications of the newly developed algorithms are under investigations.

References

[1] P. H. A. Sneath and R. R. Sokal, "Numerical Taxonomy- The Principles and Practice of Numerical Classification," W. H. Freeman, San Francisco, 1973.
[2] S. Guha, R. Rastogi, and K. Shim, "CURE: An efficient algorithm for large databases," in *Proceedings of SIGMOD* '98, Seattle, June 1998, pp. 73–84.
[3] L. Kaufman and P. J. Rousseeuw, "Clustering by means of medoids" in Statistical Data Analysis Based on the Norm," Y. Dodge, Ed., pp. 405–416. North Holland *Elsevier*, Amsterdam, 1987.
[4] L. Kaufman and P. J. Rousseeuw, "Finding Groups in Data, an Introduction to Cluster Analysis," *John Wiley &Sons*, Brussels, Belgium, 1990.
[5] R. T. Ng and J. Han, "Efficient and effective clustering methods for spatial data mining," in Proceedings of the 20th *VLDB Conference*, Santiago, Chile, Sept. 1994, pp. 144–155.
[6] News Group Dataset. Available at: www.cs.cmu.edu/afs/cs.cmu.edu.
[7] K. C. Gouda and E. Diday, "Symbolic clustering using a new similarity measure," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 20, pp. 368–377, 1992.
[8] G. D. Ramkumar and A. Swami, "Clustering data without distance functions," Bulletin of the *IEEE Computer Society Technical Committee on Data Engineering*, vol. 21, pp. 9–14, 1998.
[9] Y. El Sonbaty and M. A. Ismail, "Fuzzy clustering for symbolic data," *IEEE Transactions on Fuzzy Systems*, vol. 6, pp. 195–204, 1998.
[10] P. Bajcsy and N. Ahuja, "Location- and density-based hierarchical clustering using similarity analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, pp. 1011–1015, 1998.
[11] E. H. Ruspini, "Numerical methods for fuzzy clustering," Information Science, vol. 2, pp. 319–350, 1970.
[12] R.J. Hathaway, J.W. Devenport, and J.C. Bezdek, "Relational dual of the c-means clustering algorithms," Pattern Recognition, vol. 22, no. 2, pp. 205–212, 1989.
[13] P. Maji and S. K. Pal, "Rough-Fuzzy C-Medoids Algorithm and Selection of Bio-Basis for Amino Acid, Sequence Analysis," *IEEE Transactions on Knowledge and Data Engineering*, vol. 19, no. 6, JUNE 2007.
[14] Z.R. Yang and R. Thomson, "Bio-Basis Function Neural Network for Prediction of Protease Cleavage Sites in Proteins," *IEEE Trans.Neural Networks*, vol. 16, no. 1, pp. 263-274, 2005.
[15] D. Dubois and H. Prade, "Rough Fuzzy Sets and Fuzzy Rough Sets," Int'l J. General Systems, vol. 17, pp. 191-209, 1990.
[16] M.S. Johnson and J.P. Overington, "A Structural Basis for Sequence Comparisons: An Evaluation of Scoring Methodologies," J. Molecular Biology, vol. 233, pp. 716-738, 1993.
[17] Q. Zhang and I. Couloigner, "A New and Efficient K-Medoid Algorithm for Spatial Clustering," Gervasi *et al.* (Eds.): ICCSA 2005, LNCS 3482, pp. 181 – 189, 2005.
[18] J. C. Bezdek, "Pattern Recognition with Fuzzy Objective Function Algorithms," Plenum Press, New York, 1981.
[19] R. N. Davé and S. Sen, "Robust Fuzzy Clustering of Relational Data," *IEEE Transactions on Fuzzy Systems*, vol. 10, no. 6, 2002.
[20] D. Lewis. Reuters-21578 Text Categorization Test Collection. Available http://www.research.att.com.
[21] Dae-Won Kim, Kwang H. Lee and Doheon Lee, "Fuzzy cluster validation index based on inter-cluster proximity," *Pattern Recognition Letters* 24 (2003) 2561–2574.
[22] R.Krishnapuram, R.Joshi, A.Nasraoui and O.Yi, "Low-complexity fuzzy relational clustering algorithms for Web mining," *Fuzzy Systems, IEEE Transactions*, vol.9, pp. 595--607, Aug 2001.
[23] M. Ester, H. Kriegel, and X. Xu, "Knowledge discovery in large spatial databases: focusing techniques for efficient class identification," International Symposium on Advances in Spatial Databases, vol. 951. Springer, Portland, ME, pp. 67–82, 1995.
[24] M. Camila, N. Barioni, L. Razente, J.M. Traina, and C. Traina, "Accelerating k-medoid-based algorithms through metric access methods," Journal of Systems and Software vol. 81, pp. 343–355, 2008.
[25] R. J. Hathaway and J. C. Bedeck, "NERF c-means: Non-Euclidean relational fuzzy clustering," *Pattern Recognition*, vol. 27, pp. 429–437, 1994.
[26] M. A. Ismail "Soft Clustering: Algorithms and validity of solutions," *Fuzzy Computing*. Amsterdam, the Netherlands: Elsevier, 1988, pp. 445-471.