International Journal of Data Mining Techniques and Applications
Volume: 03 Issue: 02 December 2013, Page No.81- 88
ISSN: 2278-2419

Robust Clustering Algorithm Based on Complete Link Applied to Selection of Bio-Basis for Amino Acid Sequence Analysis

Mohamed A. Mahfouz

Department of Computer Engineering, AIET, Alexandria, Egypt

Abstract-Robust Clustering methods are aimed at avoiding unsatisfactory results resulting from the presence of certainamount of outlying observations in the input data of many practical applications such as biological sequences analysis or gene expressions analysis. This paper presents an algorithm termed as rough possibilistic complete link clustering algorithm (RPLINK) that maximizes the average similarity between pairs of patterns within the same cluster and at the same time the size of a cluster is maximized by computing the zeros of the derivative of appropriate objective function. The proposed algorithm is comprised of a judicious integration of the principles of rough sets, Complete Link and possibilistic clustering paradigm. This integration enables robust selection of the minimum set of the most informative bio-bases from the lower bound of the produced clusters. RPLINK along with the proposed initialization procedure with a few less sensitive input parameters shows a high outliers rejection capability as it makes their membership very low furthermore it does not requires the number of clusters to be known in advance and it can discover clusters of non convex shape. The effectiveness and robustness of the proposed algorithms, along with a comparison with other algorithms, have been demonstrated on different types of protein data sets.

Index Terms—Data Mining, Fuzzy Clustering, Relational Clustering, Hierarchical Clustering**,** Rough Sets, Bioinformatics.

I. INTRODUCTION

Cluster Analysis methods aim to detect homogeneous groups with large separation amongthem [25], but thesearch of groups or clusters can be completely blemished due to the lack of robustnessof standard clustering methods. For instance,several very large groups canappear artificially joined together as constituting a single cluster, or some spuriousclusters made up merely of outlying observations can be detected. Data Mining inlarge, high dimensional data sets [26] are most likely to have such troubles intheir "unsupervisedlearning" step. Another reason for the importance of Robust Clustering techniques relies on the strong relation between Cluster Analysis and Robust Statistics that have already been pointed out [27].

Identifying sets of homologous proteins in a protein database using clustering techniques allows further analysis of such huge data such as protein family discovery, function prediction, and database compression. Protein sequences that are very much alike or similar may probably have a similar biochemical function or three dimensional structures. In biological sequences, the only available information is the numerical values that represent the degrees to which pairs of sequences in the data set are related. Algorithms that generate partitions of that type of relational data are usually referred to as relational or pairwise clustering algorithms. Therefore, the relational clustering algorithms can be used to cluster biological subsequences if one can come up with a similarity measure between the pairs of subsequences. The pairwise similarities are usually stored in the form of a matrix called the similarity matrix.Medoid-based algorithms like PAM and CLARA [3], [4] respectively are examples of relational clustering algorithms in which a cluster is represented by the most centrally locatedobject in the cluster (instead of cluster centre as in centeroid based algorithms) as its representativewhich makes medoid-based algorithms applicable to relational data. 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 iscomputed, and the best set of medoids is retained. Further progress is associated the algorithm CLARANS (Clustering Large Applications based upon RANdomized Search) [5]. 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. If a neighbor represents a better partition, the process continues with this new node.

Hierarchical Clustering algorithms introduced in [1]suffer from their inability to perform adjustment once a merge or a split decision is performed i.e. bad decisions taken at some step lead to low-quality clusters. Also traditional Hierarchical clustering algorithms suffers from high computational complexity. Furthermore, hierarchical algorithms that rely on distance measures when deciding if merge or split clusters usually perform well only on clusters with spherical shapes. Othervariation of hierarchical clustering methods that try to tackle these problems rely on either clusters proximity or clusters interconnectivity or both [2]. In [29] authors proposed an iterative hard clustering algorithm based on complete link. Recent scalable hierarchical algorithms based on single linkage strategy are found in[31].PAM, CLARA, CLARANS, Hierarchical Clustering andother early algorithms for relational clustering as in [7]-[9], [29]generate crisp clusters. When the clusters overlap as the case in sequence clustering, we may desire fuzzy clusters.Some of the early fuzzy relational clustering algorithms are introduced in [11], [12] and [19]. The Relational Fuzzy C-Means (RFCM) [12] is extended in [21] to ease the restrictions that RFCM imposes on the dissimilarity matrix. More recently this approach is generalizedfurther by including an extension to handle datasets containing noise and outliers [19].The study most relevant to our focus here is [13],[22], [28], [29].In [22] a fuzzy clustering for a relational data termed as FCMdd(Fuzzy C-Medoids) is proposed and

International Journal of Data Mining Techniques and Applications
Volume: 03 Issue: 02 December 2013, Page No.81- 88
ISSN: 2278-2419

compared with the Relational Fuzzy C-Means algorithm (RFCM) and found to be more efficient. In [13] the principles of rough sets, fuzzy sets [15] is applied to both the hard and fuzzy c-medoids algorithm [22] and rough-fuzzy c-medoids algorithm is proposed to select the most informative bio-bases [14]. In [13] the amino acid mutation matrix [16] is used in computing the similarity between objects (sequences).In [28] authors uses randomized search along with soft clustering[23] to reduce the complexity of the rough fuzzy c-medoids algorithms[13].

However both centeroid-based ormedoid-based algorithms can only discover clusters of convex shape, sensitive to outliers, and require the number of clusters to be given as input. Fixing the number of clusters represents a severe biasthat affects, and sometimes prevents, the extraction of useful knowledge from data.To deal with this problems a robust clustering algorithm termed as rough possibilistic complete link clustering (RPLINK) is proposed. In RPLINK the use of cluster centeroids [13] is avoided instead the pairwise average similarity between objects in a cluster is used that makes the produced cluster boundaries are not forced to have a pre-specified geometric shape depending on the used similarity measure as the case of medoids or centeroids based algorithms.RPLINK shows a high outliers rejection capability as it makes their membership very low.

An initialization procedure is proposed that does not require several input parameters as in [13] instead it needs only one parameter (a threshold on the average similarity within a cluster) that can be systematically estimated as described in section III. Also specifying a threshold on the average similarity within a cluster is easy for a user to understand. The proposed initialization procedure are able to identify candidate outliers that are initially excluded from the computation done in the possibilistic memberships computation phase. A good initial set of clusters must be found before applying the rough possibilistic algorithm which works as a refinement step. The rest of the paper is organized as follows: section II; reviews related concepts and algorithms. Section III; describes the proposed algorithms. Section IV; compares the performance of the proposed algorithms to several related algorithms. Finally section V; concludes the paper with summary and ideas for future work.

II. BACKGROUND

A. Possibilistic Clustering Paradigm

The possibilistic approach to clustering proposed by Keller and Krishnapuram [34], [35] assumes that the membership function of a data point in a fuzzyset (or cluster) is absolute, i.e. it is an evaluation of a degree of typicality not depending on the membership values of the same point in other clusters.Let $X = \{x_1, x_2 \ldots..,x_n\}$ be a set of n unlabeled data points, $Y = \{y_1, y_2 \ldots\ldots ; y_c\}$ a set of c cluster centers (or prototypes) and $U = [u_{pq}]$ the fuzzy membership matrix. In the Possibilistic C-Means (PCM) algorithms the constraints on the elements of Uare relaxed to:

$$u_{pq} \in [0,1] \qquad \forall\ p, q \quad 0 < \sum_{q=1`}^{n} u_{pq} < n \qquad \forall\ p$$

$$\bigvee_{p} u_{pq} > 0 \qquad \forall\ q$$

These requirements simply imply that a cluster cannot be empty and each pattern must be assigned to at least one cluster. This turns a standard fuzzy clustering procedure into a mode seeking algorithm [34]-[35]. The objective function contains two terms; the first one is the objective function of the fuzzy *C*-Means [30], while the second is a penalty term considering the entropy of clusters as well as their overall membership values:

$$J_m(U,Y) = \sum_{p=1}^{s}\sum_{q=1}^{r} u_{pq} E_{pq} + \sum_{p=1}^{s} \frac{1}{\beta_p} \sum_{q=1}^{r} (u_{pq} \log u_{pq} - u_{pq}) \quad (1)$$

where $E_{pq} = \left\| x_p - y_p \right\|^2$ the squared Euclidean distance, and the parameter β_p should be estimated before the clustering procedure starts depending on the average size of the *p-th* cluster. The solution that is obtained by minimizing the above objective function will be highly dependent on the parameter β_p. Note that if $B_p \to \infty \quad \forall\ p$ (i.e., the second term of$J_m(U,Y)$ is omitted), then a trivial solution is obtained by the minimization of the remaining cost function (i.e., $u_{pq}=0 \ \forall\ p, q$, as no probabilistic constraint is assumed).The pair $(U; Y)$ minimizes J_m, under the above constraints only if :

$$u_{pq} = e^{-E_{pq}/\beta_p} \qquad \forall\ p, q \quad (2)$$

and $$y_p = \sum_{q=1}^{r} x_q u_{pq} / \sum_{q=1}^{r} u_{pq} \qquad \forall\ p. \quad (3)$$

Equations (2) and (3) can be used as formulas for recalculating the membership functions and the cluster centers.

B. Rough Clustering

A rough cluster is defined in a similar manner to a rough set [15] that it has a lower and upper approximation. The lower approximation of a rough cluster contains objects that only belong to that cluster. The upper approximation of a rough cluster contains objects in the cluster which are alsomembers of other clusters. An important distinction between rough clustering and other conventional clustering approaches is that, with rough clustering, an object can belong to more than one cluster thereby allowing overlapping of clusters. A appropriate distance measure should be used in rough clustering such that the strict requirements of in-discernibility relation used in normal clustering is relaxed. Thus rough clustering allows for grouping of objects based on a notion of similarity relation rather than based on equivalence relation.

C. Similarity Measures

The similarity score between two subsequences $x_j$and v_i that is used in the experiments of [13] is the non-gapped pair-wise homology alignment score, $h(x_i, v_j)$ and is defined as:

$$h(x_i, v_j) = \sum_{k=1}^{d} M(x_{ik}, v_{jk}) \quad (4)$$ Where *d*is the number of characters in the subsequences and is set to 8 in the preprocessing step of [13]. M(x_{ik},v_{jk}) is a homology alignment score (similarity value) can be obtained from a table lookup called mutation matrix. A mutation matrix has 20 columns and 20 rows. M(n,m) is the value at the n^{th} row and m^{th} column of the mutation matrix and it is integer value that represents the probability or a likelihood value that the nth amino acid mutates to the m^{th} amino acid after a particular evolutionary

International Journal of Data Mining Techniques and Applications
Volume: 03 Issue: 02 December 2013, Page No.81- 88
ISSN: 2278-2419

time. Each character in a subsequence corresponds to row/column in the mutation matrix. Also the mutation matrix is asymmetric i.e. $h(x_i, v_j) \neq h(v_j, x_i)$. Another quantitative measure to evaluate the similarity between two subsequences in terms of nongapped pairwise homology alignment score is also reported in [13] as follow:

$$DOR(x_i, v_j) = h(x_i, v_j) / h(v_j, v_j) \quad (5)$$

It is the ratio between the non-gapped pair-wise homology alignment scores of two input subsequences x_i and v_j based on an amino acid mutation matrix to the maximum homology alignment score of the subsequence v_j. It is used to quantify the similarity in terms of homology alignment score between pairs of subsequences. If the functions of two subsequences are different, the DOR between them is small. A high value of DOR(x_i,v_j) between two subsequences x_i and v_j asserts that they may have the same function statistically. If two subsequences are same, the DOR between them is maximum, that is, DOR(x_i,v_j)= 1. Thus,

- Dor(x,x)=1
- 0 <DOR(x_i,v_j)≤1.
- DOR(x_i,v_j)≠ DOR(v_j, x_i).

The similarity matrix using either h or Doris actually a complete one not an upper triangular one. While the concept of DOR[13] are used in deriving a method for selecting initial bio-bases (medoids), the non-gapped pair-wise homology alignment score h(x_i,v_j) is used in the computation involved in the clustering process and in evaluating the performance. Several recent efficient approaches are based on partial matching of sequences instead of all-versus-all comparison of sequences such as [17] and [24]. The proposed algorithm is applicable to both partial or complete computation of similarity between sequences . The similarity measure that is used here is based on all-versus-all comparison of sequences to allow comparison with the most relevant approach [13].

III. METHODS

The main idea of the proposed algorithm is to avoid the use of cluster centers which makes the cluster boundaries take a pre-specified geometric shape depending on the used similarity measure. The proposed solution of the problem resides in modifying the objective function in equ. (1) so that it minimizes (maximizes) the average distance (similarity) between every pair of data points within the cluster. Based on equ.(1) The objective function to be maximized is

$$J_m(U) = \sum_{p=1}^{k} \left(\frac{1}{\sum_{i=1}^{n}\sum_{j\neq i}^{n} u_{pi}\, u_{pj}} \sum_{i=1}^{n}\sum_{j\neq i}^{n} u_{pi}\, u_{pj}\, S(x_i, x_j) \right) - \beta \sum_{i=1}^{n} (u_{pi} \ln u_{pi} - u_{pi}) \quad (6)$$

Where S(x_i,x_j) is the similarity between subsequence *i* and subsequence *j* and represents an entry (*i*,*j*) in the similarity matrix S. And $u_{pi}$is the possibilistic membership of subsequence *i* in cluster p and represents an entry (*i*,*p*) in the membership matrix M.

$$\frac{\partial J_m}{\partial u_{pi}} = \frac{\sum_{j\neq i}^{n} u_{pj} S(x_i, x_j)}{\sum_{i=1}^{n}\sum_{j\neq i}^{n} u_{pi} u_{pj}} - \frac{\sum_{j\neq i}^{n} u_{pj}}{\left(\sum_{i=1}^{n}\sum_{j\neq i}^{n} u_{pi} u_{pj}\right)^2} \sum_{i=1}^{n}\sum_{j\neq i}^{n} u_{pi} u_{pj} S(x_i, x_j) \quad (7)$$

$$- \beta \ln(u_{pi}) - 1 + 1 = 0$$

The solution for equation (7) is

$$u_{pi} = e^{-E_{pi}/\beta} \qquad \forall\ p, i \quad (8)$$

where

$$E_{pi} = \frac{\sum_{j\neq i}^{n} u_{pj}}{\left(\sum_{i=1}^{n}\sum_{j\neq i}^{n} u_{pi} u_{pj}\right)^2} \sum_{i=1}^{n}\sum_{j\neq i}^{n} u_{pi} u_{pj} S(x_i, x_j) - \frac{\sum_{j\neq i}^{n} u_{pj} S(x_i, x_j)}{\sum_{i=1}^{n}\sum_{j\neq i}^{n} u_{pi} u_{pj}} \quad (9)$$

In section III(C), equations (8) , (9) are used as formulas for recalculating the membership functions. Initial M, Sand β are given as input to the algorithm described in fig. 4.

D. The Proposed Initialization Procedure

Both the proposed algorithm and the c-medoids algorithms require pre-specifying the number of clusters in advance, which is usually unknown. Although multiple techniques exists in literature for estimating the number of clusters, they are based on running the algorithms repeatedly with different values of number of clusters and comparing the clustering results but for large data sets this will be extensive parameter fine-tuning process and totally not practical. By specifying a threshold α on the average similarity among each cluster one can control the number of initial clusters. The next section describes a systematic procedure for computing reasonable threshold α. Also the user can easily specify a value of this threshold based on the amount of homogeneity he needs in the resulting clusters. The main drawbacks of the initialization procedure in [13] is that it needs several parameters, a user cannot easily specify a suitable value for it. Actually the procedure given in (13) relies on converting the clustering problem into graph coloring problem and use simple techniques for estimating the number of colors which corresponds to initial bio-basis. Finally the use of a threshold on the average similarity allow identifying candidate outliers which will be unable to produce a cluster with acceptable size constrained by this threshold .

procedureinitproc

input :

S :Similarity matrix of size *n*×*n*

α : threshold on the average similarity (estimated in fig. 3)

prcnt: threshold on the size of acceptable cluster size

output:

U: hard membership matrix representinitial clustering

International Journal of Data Mining Techniques and Applications
Volume: 03 Issue: 02 December 2013, Page No.81- 88
ISSN: 2278-2419

c : number of clusters

begin [initproc]

1. **For** each pattern x_i, calculate the

$$Sum(x_i) = \sum_{j\neq i}^{n} S(x_i, x_j) \quad \text{for} j=1,2,...n$$

2.**Compute** $totSum = \sum_{i=1}^{n} Sum(x_i)$

3**Set***totCount*to *n*, **Set***k* to 1

4. **Add** all unlabeled patterns to cluster C_k

5. **while** (*totSum*/*totCount*< α)

begin
6.1**Remove**$x_j = \text{argmin}_{x_i \in C_k} \text{Sum}(x_i)$
6.2**Mask***$x_j$and update totSum*, Sum
6.3**Decement***totCount*
end
6. **if**(*totalCount*<*prcnt***n*)
label patterns in C_k as potential outliers, Clear C_k

label remaining patterns as potential outliers

if (no. potential outliers >avg. size of prev. C_i)

reduce α as explained in fig. 3

Go To Step 1

else

set initial memberships in U to 0 for all outliers

exit

end if

else

increment*k*

updateSum,totSum, totCount for unlabeled patterns

Go To Step 4

endif

end [initproc]

Fig. 1. The Proposed Initialization Procedure

As shown in fig. 1, The initialization procedure creates initial clusters one by one each time starts with all not previously labeled objects as elements in a candidate cluster in each iteration the object x_i that represent the minimum value in the array Sum is removed from the current cluster,the Sum(x_j)is decremented by S(x_j, x_i) for each x_j inside the candidate cluster. Finally the procedure tests every new discovered cluster if the new cluster passed a pre-specified threshold, it is declared as a new discovered cluster; otherwise the patterns within it along with remaining unlabeled patterns are marked as potential outliers. The whole procedure is repeated if the count of candidate outliers exceeds the average size of previously produced clusters.

Example: assuming α =0.75, in fig.2(a) anasymmetricsimilarity matrix(computed using Dor so the diagonal are ones) offour subsequences s_1..s_4 with the diagonal is masked and excluded from the computations and the Sum array. Fig.2(b) after masking s2 and s3. At the beginning thecandidate cluster contains all subsequences, the average similarity is (7.2/12)<0.75). After two iterations the algorithm stopped and s1 and s4 returned as intial cluster since stopped when (1.7/2)=0.85>0.75). if we continue s2,s3 marked as candidate outliers and the algorithm stops.

	s_1	s_2	s_3	s_4	Sum
s_1	-	.8	.7	.9	2.4
s_2	.6	-	.5	.1	1.2
s_3	.5	.4	-	.6	1.5
s_4	.8	.6	.7	-	2.1
-	-	-	-	-	7.2

(a)

	s_1	s_2	s_3	s_4	sum
s_1	-	-	-	.9	0.9
s_2	-	-	**.5**	-	-
s_3	-	**.4**	-	-	-
s_4	.8	-	-	-	0.8
-	-	-	-	-	1.7

(b)

Fig 2. (a) the initial sum array (b) sum after removing s2,s3

A. Tuning The Input Parameters α and β

The following is a systematic procedure that can be followed for computing a range for suitable value for the input parameter α and β for a given dataset.

Procedure EstimateAlphaBeta

input:

Seq: is the whole input sequence (string of alphabetic)

output:

S : Similarity matrix of size *n*×*n*

α : threshold on the average similarity

β :parametersof the objective function in equ. (6)

begin [EstimateAlphaBeta]

1. Compute the Similarity matrix S using Dor
2. Select randomly very large number (hundreds of thousands) of subsets of input subsequences.
3. Compute 1000 bins histogram for the average similaritiesfor each subset selected by step 2.
4. Get the number of first bins fbins such that the sum of the count of them is near 0.001 of the total count.
5. Compute α = min. value + fbins * step (a value close to the lower end of the range of the average similarities).
6. Compute average E_{pi} for random sample of objects in the identified subsets in step 5 (counted in the lower end)
7. Compute β using equ. (10)

end [EstimateAlphaBeta]

Fig. 3. The proposed procedure for estimating α and β

By Identifying subsets that having average similarities less than the computed threshold in step 4, a systematic approach for estimating a suitable value for β is to use the average E_{pi} for random sample of objects in the identified subsets and assuming 0.5 membership for such value , using equ. (6), a suitable value for β is:

$$\beta = \frac{-\ln 0.5}{average \quad E_{pi}} \qquad (10)$$

International Journal of Data Mining Techniques and Applications
Volume: 03 Issue: 02 December 2013, Page No.81- 88
ISSN: 2278-2419

B. The Proposed Algorithm

After computing the initial hard membership matrix U and the number of clusters c as in fig. 1, and estimating a suitable value for β as in fig. 3. The refinement step starts as shown in fig. 4. The algorithm iterates over all the objects and computes new memberships using equ. (8) and (9).

Input:

*S/*Similarity matrix of size n×n*/*

*β /*the parameters of the objective function in equ. (6) */*

ε / the threshold controlling the convergence*/*

Ϭ / the threshold used to identify border objects*/*

c / the number of clusters to be found*/*

w / weight for border objects*/*

*U /*initial hard membership matrix produced as in fig. 1*/*

Output:

U / matrix of possibilistic memberships*/*

Begin [RPLINK]

Repeat

store Memberships *U* in U'

for each object x_i

begin

compute E_{pi} for *p* = 1,2,3...*cusing equ. (7)*

if (E_{pi} is minimum) // x_i may $\in \underline{B}(C_p)$

$u_{pi} = e^{-E_{pi}/\beta}$

for each(q≠p)

if((u_{pi} - u_{qi})< Ϭ)) // $x \in \bar{B}(C_p), x \in \bar{B}(C_q), x \notin \underline{B}(C_p)$

begin $u_{qi} = w\ e^{-E_{qi}/\beta}$; $u_{pi} = w\ e^{-E_{pi}/\beta}$ **end**

else

u_{qi}=0

end if

endif

end

Until ($\| U - U' \| < \varepsilon$)

Output *U*

End [RPLINK]

Fig. 4.Rough Possibilistic Complete Link Clustering Algorithm (RPLINK)

Each time the new memberships are computed for an object x_i in all clusters the maximum membership (which corresponds to minimum E_{pi}) are compared to the next values if the difference between the max memberships and any of the next values u_{qi} less than experimentally tuned threshold Ϭ (equal to 0.15) the object x_i are considered border object for cluster *p* and *q* and the memberships are modified by an input weight *w*(set to value 0.7as in [13]) . If no such memberships exists in thenext values then x_i is considered core object in cluster p and its membership is left as it is. Memberships in other clusters in which x_i is neither core or border object are set to 0. Finally, the algorithm terminates if no significant change in memberships which is decided by the input parameter ε .

IV. RESULTS AND DISCUSSIONS

The performance of RPLINK is compared to the following algorithms:

1) C-medoids(RFCMdd, FCMdd, HCMdd and RCMdd [13])
2) Neural Network(MI)[14]
3) Genetic algorithm(GAFR) [20]

C-medoids algorithms are re-implemented in C# as RPLINK to allow comparing execution time and to allow using different performance indices such as Silhouette width[33] and Dun index[32]. In comparing with GAFR and MI the values reported in [13] are used. The implemented programs are run in windows7 environment having a machine configuration of*Pentium IV, 3.2 GHz, 1 Mbyte cache, and 2 GB of RAM*

A. Datasets and Preprocessing

To analyze the performance of the proposed algorithms while reducing the risk that our conclusions might be valid only on a particular corpus, all the five HIV datasets that are reported in [13] are used. Each dataset is a sequence of characters from the set {A,C,D,E,F,G,H,I,K,L,M,N,P,Q,R,S,T,V,W,Y}. NP_057849 and NP_057850represents the longest and the shortest sequence among the five datasets and have length of 1435 and 500 characters respectively.The subsequences are obtained from the protein sequences through moving a sliding window with eight residues. The total number of subsequences with eight residues in NP_057849 and NP_057850 are 1428, and 493 respectively.

B. Quality Measures

Several techniques assess both intra-cluster homogeneity and inter-cluster separation, andcompute a final score as the linear or non-linear combination of the two measures. The two quality measures β and γ used in [13] are defined as follows

$$\beta = (1/c)\sum_{i=1}^{c}(1/n_c)\sum_{j=1}^{n} DOR\ (x_j, v_i) \quad (11)$$

$$\gamma = \max_{i,j}\frac{1}{2}(DOR\ (v_j, v_i) + DOR\ (v_i, v_j)) \quad (12)$$

In the performance analysis of the proposed algorithm, the Dunn Index and Silhouette width are used. Dunn Index [32] is a verywell-known example of non-linear combinations that measures the ratio between the largest cluster similarity and the smallest intra-cluster similarityin a partitioning. It is defined as

$$D(C) = \max_{C_k \in C} \in \left(\max_{C_l \in C} \frac{\text{sim}(C_k, C_l)}{\min_{C_m \in C} \text{diam}(C_m)}\right) \quad (13)$$

where diam(C_m) is the minimum intra-cluster similarity within cluster C_m, and sim(C_k,C_l) isthe maximal similarity between pairs of data items *i* and *j* with $i \in C_k$ and$j \in C_l$. TheDunnIndex is limited to the interval [0,+∞] and should be maximized.Thesilhouette value [33] for an individual data item represents a confidence indicator on its membershipin a specific cluster. The Silhouette Width for a partitioning is computed as the average silhouette value over all data items.

International Journal of Data Mining Techniques and Applications
Volume: 03 Issue: 02 December 2013, Page No.81- 88
ISSN: 2278-2419

A Global Silhouettevalue, GS_u, can be used as an effective validity index for U.

$$GS_u = \frac{1}{k}\sum_{j=1}^{k}\sum_{x_i \in C_j} s_i \qquad (14)$$

wheres_iThe Silhouette value for the i-th sample x_i in a cluster C_j and is defined as follows:

$$s_i = (b_i - a_i)/\max(b_i, a_i) \qquad (15)$$

wherea_i denotes the average distance between i and all data items in the same cluster, and b_i denotes the average distance between i and all data items in the closest other cluster (which is defined as the one yielding the minimal b_i). When $s_i$is close to 1, one may infer that the i-th sample x_i has been well clustered.When $s_i$is close to zero, it suggests that the i-th sample could also be assigned to the nearest neighboring cluster.Furthermore, it has been demonstrated that the previous equation can be applied to estimatethe most appropriate number of clusters for U. In this case the partition with themaximum $S_u$is taken as the optimal partition.

V. PERFORMANCE EVALUATION

In the following experiments, for each chosen number of clusters, the results of 20 runs are averaged to represent the results of the corresponding algorithms on the selected number of clusters. The initial procedure proposed in this paper is used with RPLINK while the initial procedure of [13] is used in comparing with C-Medoids.

A. Comparison using the values of β, γ reported in [13]

In Table 1, the values for the other algorithms are those reported in [13]. All the reported results in table 2 from [13] are produced by initializing the algorithms with c bio-bases that are generated using the methods proposed by Berry et al. (GAFR) and Yang and Thomson (MI) while RPLINK is initialized using the proposed procedure.

Table 1

Performance of RPLINK

compared to all above listed Algorithms on NP_057849

Algorithm	Param.	B	γ	Param.	β	γ
RPLINK	C=13	0.792	0.853	C=27	0.826	0.799
RFCMdd		0.736	0.914		0.801	0.819
FCMdd		0.719	0.914		0.746	0.828
RCMdd		0.612	0.938		0.635	0.829
HCMdd		0.607	0.938		0.621	0.827
MI		0.611	0.944		0.625	0.913
GAFR		0.609	0.962		0.618	0.902
RPLINK	C=26	0.806	0.805	C=36	0.891	0.672
RFCMdd		0.801	0.821		0.836	0.681
FCMdd		0.746	0.837		0.767	0.701
RCMdd		0.632	0.836		0.651	0.751
HCMdd		0.618	0.844		0.643	0.751
MI		0.624	0.913		0.637	0.854
GAFR		0.616	0.902		0.646	0.872

A 20 runs were executed to get a reliable average measure of the validation indices for RPLINK at various numbers of clusters. The value of the a threshold for our proposed initialization algorithm was also varied to produce the same number of clusters as the other algorithms at the points of comparison. The value of any index at any cluster number that wasn't feasible to generate by varying a threshold, was generated using linear interpolation. The proposed algorithm scored high values for both β, γ and proved to be superior to other tested algorithms. The gain in β was higher than for γ.

B. Comparison with C-Medoids Algorithm using Other Performance Indices

In order to compare RPLINK with C-Medoids on different performance indices, the algorithms are re-implemented and initialized using the initial procedure used in [13]. As shown in table 2, RPLINK scored high values for both Dunn Index and Silhouette Width and proved to be superior to other tested algorithms. the gain in Silhouette Width is higher than for Dun index because the proposed objective function match Silhouette Width more than Dun index, or in other words the proposed algorithm achieve higher compactness than separation.

Table 2

Performance of RPLINK compared to Several C-Medoids Algorithms using other performance indicesonNP_057849 using c=36

Algorithm	Dunn Index	Silhouette Width
RPLINK	0.321574	0.362815
RFCMdd	0.305913	0.280974
FCMdd	0.291722	0.289233
RCMdd	0.264625	0.228046
HCMdd	0.24615	0.211839

Runtime Analysis

The results in table 3 represents the average execution time for RPLINK and RFCMddIn this experiment the corresponding algorithms when they are applied to the dataset NP_057849 and NP_057850 for different number of clusters.The runtime in Table 3.It is clear that RPLINK is slightly slower than RFCMdd on NP_057849 and comparable to it for the smaller

International Journal of Data Mining Techniques and Applications
Volume: 03 Issue: 02 December 2013, Page No.81- 88
ISSN: 2278-2419

dataset NP_057850. Also the higher the cluster count the higher the execution time for both algorithms.

Table 3

Execution Time in (ms) for different number of clusterscompared to RFCMdd on NP_057849 and NP_057850

Cluster Count	NP_057849		NP_057850	
	RPLINK	RFCMdd	RPLINK	RFCMdd
6	04122	05213	0771	0882
13	14223	13612	1233	1423
26	28312	24063	3677	3114
36	40802	32713	4122	3821
50	61507	44329	7231	5754

Table 4 compare the average execution time for RPLINK compared to different C-Medoids algorithms on NP_057849 the results shows that that RPLINK as FCMdd and RFCMdd is much slower than the hard algorithms. Even though the computation of the complete link is expected to be much higher than the computation of the objective function of the c-medoids but It was also noted through the experimental results that it has a very high convergence rate, in addition to the enhanced quality of the clustering results.

Table 4

Execution Time in (ms) Compared to Different C-Medoids Algorithmsusing c=36

Algorithm	NP_057849
RPLINK	40802
RFCMdd	32713
FCMdd	240834
RCMdd	160563
HCMdd	4379

VI. CONCLUSION AND FUTURE WORK

This paper presented a novel robust clustering algorithm along with an initialization procedure in which a threshold on the average similarityrather than a pre-specified number of clusters is specified.By applyingthe proposed algorithm inbiological sequences analysis and comparing itsresults with the results obtained for other classical and state-of the-art clustering algorithms, the proposed clusteringalgorithm showed remarkable performance and proved tobe competitive to other widely used algorithms forbiological sequence clustering. The following can be concluded from the analysis of the algorithms and the experimental results:

- The proposed possibilistic approach is able to deal with outliers and produce higher quality of results thanC-Medoids algorithms in terms of both Silhouette width and Dun Index.
- The gain in Silhouette width is higher than of Dun index as it matches the proposed objective function.
- The algorithm RPLINK is compared to GAFR and MI in terms of quality measured as β, γand shows highly competitive results.
- By keeping the rows and columns basis of the membership matrix, RPLINK isslightlyslower thanRFCMdd.
- RPLINK needs less parameters than C-Medoids algorithms and the few parameters that are needed can be systematically computed or fine tuned.

The following are possible future direction for research:

- Using single linkage strategy instead of complete link is under investigation.
- Although the methodology of integrating the principles of fuzzy sets, semi-fuzzy or soft clustering models and c-medoids algorithm has been demonstrated for biological sequence analysis, the concept can be applied to other relational unsupervised classification problems.

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] George Karypis, Eui-Hong (Sam) Han, and Vipin Kumar, "Chameleon: A hierarchical clustering using dynamic modeling,". Computer, 32(8): pp. 68–75, 1999.
[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] Protein Sequence Datasets. Available: http://www.ncbi.nlm.nih.gov.
[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] 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.
[10] Y. El Sonbaty and M. A. Ismail, "Fuzzy clustering for symbolic data," *IEEE Transactions on Fuzzy Systems*, vol. 6, pp. 195–204, 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, 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] P. A. Vijaya, M. NarasimhaMurty and D. K. Subramanian "Efficient median based clustering and classification techniques for protein sequences," Pattern Analysis and Applications, vol . 9, pp. 243–255, 2006.
[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] E.A. Berry, A.R. Dalby, and Z.R. Yang, "Reduced Bio-Basis Function Neural Network for Identification of Protein Phosphorylation Sites: Comparison with Pattern Recognition Algorithms," Biology and Chemistry, vol. 28, pp. 75-85, 2004.

[21] R. J. Hathaway and J. C. Bedeck, "NERF c-means: Non-Euclidean relational fuzzy clustering," *Pattern Recognition*, vol. 27, pp. 429–437, 1994.
[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. A. Ismail "Soft Clustering: Algorithms and validity of solutions," Fuzzy Computing. Amsterdam, the Netherlands: Elsevier, 1988, pp. 445-471.
[24] S. Hazelhurst and Z. Lipták" KABOOM! A new suffix array basedalgorithm for clustering expressionon data," Bioinformatics, vol 27, pp. 3348-3355, 2011.
[25] Kaufman L and Rousseeuw PJ "Finding groups in data: an introduction to cluster analysis,". Wiley, New York, 1990.
[26] Croux C, Gallopoulos E, Van AelstS andZha H "Machine learning and robust data mining," Computer Statistics and Data Analysis, vol. 52, pp.151–154, 2007.
[27] Schynsa M, Haesbroeck G, CritchleyF RelaxMCD "smooth optimization for the minimum covariance determinant estimator," Computer Statistics and Data Analysis, vol. 54, pp. 843–857, 2010.
[28] M. A. Mahfouz, M. A. Ismail, "Efficient Soft Relational Clustering based on Randomized Search Applied to Selection of Bio-Basis for Amino Acid Sequence Analysis," The Proceedings of the International IEEE Conference on Computer Engineering&Systems , Ain Shams, EGYPT, pp. 287-292, Nov 2012.
[29] H. ShararaM.A.Ismail,Biosoft: "αCORR: A novel algorithm for clustering gene expression data," Bioinformatics and Bioengineering,
2007. BIBE 2007. Proceedings of the 7th IEEE International Conference, pp. 974–981, 2007.
[30] A. Baraldi, P. Blonda "A survey of fuzzy clustering algorithms for pattern recognition.systems," Man, and Cybernetics, Part B, IEEE Transactions, vol. 29, no. 6, pp. 778 – 785, 1999.
[31] Ling-Hong Hung, and Ram Samudrala "fast_protein_cluster: parallel and optimized clustering oflargescale protein modeling data," Bioinformatics, 2014.
[32] J. C. Dunn, "Well separated clusters and fuzzy partitions", Journal of Cybernetics,4 (95 − 104), 1974.
[33] P. J. Rousseeuw, "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis", Journal of Computational and Applied Mathematics, vol. 20, pp. 53 − 65, 1987.
[34] R. Krishnapuram, J. M. Keller "A possibilistic approach to clustering," Fuzzy Systems, IEEE Transactions on 1(2) (1993) 98–110.
[35] R. Krishnapuram, J. M. Keller, "The possibilistic c-means algorithm: insights and recommendations," Fuzzy Systems, IEEE Transactions, 1996, pp. 385–393.