Journal of Mathematics and System Science 6 (2016) 205-214
doi: 10.17265/2159-5291/2016.05.005

AVLINK: Robust Clustering Algorithm based on Average Link Applied to Protein Sequence Analysis

Mohamed A. Mahfouz, PhD

Department of Computer and Systems Engineering, Faculty of Engineering, Alexandria University.

Received: March 01, 2016 / Accepted: April 02, 2016 / Published: May 25, 2016.

Abstract: Robust Clustering methods are aimed at avoiding unsatisfactory results resulting from the presence of certain amount of outlying observations in the input data of many practical applications such as biological sequences analysis or gene expressions analysis. This paper presents a fuzzy clustering algorithm based on average link and possibilistic clustering paradigm termed as AVLINK. It minimizes the average dissimilarity 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 proposed objective function. AVLINK along with the proposed initialization procedure show 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 have been demonstrated on different types of protein data sets.

Key words: Data Mining, Fuzzy Clustering, Relational Clustering, Hierarchical Clustering, Bioinformatics.

1. Introduction

Clustering is a data mining task aims to divide input objects into groups (clusters) with high similarity between objects inside each group and at the same time with large separation among the groups themselves [25]. The search for such groups is highly affected by the existing of noise in the input data [27]. For instance, several very large groups can appear as a single cluster, or several clusters made up merely of outlying observations can be detected. Data mining in large, high dimensional datasets [26] are most likely to have such troubles in their clustering step.

Clustering of huge data in protein database allows further analysis of such data such as discovering protein families, and predicting new function, and compressing database. Similar protein sequences may probably have a similar biochemical function or structure. Biological sequences datasets are not numerical but represented as a sequence of characters and the only available information is the similarity between pair of characters. Therefore, only relational clustering algorithms can be used to cluster biological subsequences a similarity matrix between subsequences.

Medoid-based algorithms such as PAM [3] and CLARA [4] or CLARANS [5] are examples of relational clustering algorithms in which a cluster is represented by the most centrally located object in the cluster as its representative (instead of cluster centre as in centroid based algorithms such as C-Means [1]). Although there are a variation of both centroid- based algorithms and medoid-based algorithms, however the objective function of all of them to be minimized in each step, is a Least Squares type objective function. By minimizing this type of objective function, only clusters of convex shape can be discovered and a single outlier object may lead to very bad clustering results. Also they require the number of clusters to be given as input.

Hierarchical clustering on the other hand does not require the number of clusters to be known in advance.

Corresponding author: Mohamed A. Mahfouz, Ph.D., Department of Computer and System Engineering, Faculty of Engineering, Alexandria University.

They are divided into two category agglomerative methods, which progressively merge objects according to their degree of similarity, and divisive methods, start with the whole dataset as one cluster and progressively subdivide the data set [1]. Several linkage criteria are used such as single linkage, complete linkage or average linkage.

Although Hierarchical Clustering algorithms does not require the number of clusters as input, it suffer from their inability to overcome early bad decisions while building the Hierarchy of clusters as a representation of the data. Also traditional Hierarchical clustering algorithms suffers from high computational cost. Also they work well only on clusters with spherical shapes. Other variations 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 average link. Recent scalable hierarchical algorithms based on single linkage strategy are found in [31].

PAM, CLARA, CLARANS, Hierarchical Clustering and other 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 [18-19]. The Relational Fuzzy C-Means (RFCM) [12] is extended in [21] to release the restrictions that RFCM requires on the dissimilarity matrix. More robust approach is found in [19].

The study most relevant to our focus here is [13] ,[22], [28] and [29]. In [22] a fuzzy clustering for a relational data termed as FCMdd (Fuzzy C-Medoids) is proposed and 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]. The amino acid mutation matrix [16] is used in computing the similarity matrix of subsequences in [13]. 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 centroid-based or medoid-based algorithms can only discover clusters of convex shape, sensitive to outliers, and require the number of clusters to be given as input. In AVLINK the use of cluster centeroids or medoids is avoided instead the pairwise average dissimilarity between objects in a cluster is used that makes the produced cluster boundaries are not forced to have a pre-specified geometric shape.

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 later. Also specifying a threshold on the average similarity within a cluster is easy for a user to understand. The proposed initialization procedure is 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 AVLINK which works as a refinement step.

The rest of the paper is organized as follows: section 2; reviews related work and describes the proposed algorithms along with the initialization procedure. Section 3; compares the performance of AVLINK to several related algorithms. Finally section 4; concludes the paper with summary.

2. The Proposed Approach

The aim of this research study is to develop a possibilistic clustering technique that is applicable to relational data such as protein sequence data. In the following sections the related possibilistic c-means (p-cmeans) is explained followed by the proposed objective function in which the objective function of the p-cmeans is modified to avoid the use of cluster centers and to minimizes the average dissimilarity between every pair of data points within the cluster instead. Possibilistic clustering algorithm works as a refinement step [17] and requires a good initial clustering before they are applied. A proposed initialization procedure which needs a single automatically tuned parameters is presented along with AVLINK. Finally the proposed dissimilarity measures to be used from the domain of sequence analysis are presented.

2.1 Related Work

The possibilistic approach to clustering [17] and [24] assumes that the membership function of a data point in a cluster is an evaluation of a degree of typicality and does not depend 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, $Y = \{y_1, y_2 \ldots\ldots ; y_k\}$ a set of k cluster centers (or prototypes) and U the fuzzy membership matrix. The constraints on the elements of U are relaxed to:

1) $u_{pq} \in [0,1] \quad \forall p, q$

2) $0 < \sum_{q=1`}^{n} u_{pq} < n \qquad \forall p$

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

These requirements simply imply that a cluster size cannot equal zero and each pattern should belong to at least one cluster.

Equ. (1) represents The objective function of p-cmeans. It 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 sum of the entropy of clusters minus their overall membership values:

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

Where $\mathrm{E}_{\mathrm{pq}} = \left\| \mathrm{x}_{\mathrm{p}} - \mathrm{y}_{\mathrm{p}} \right\|^2$ the squared Euclidean distance, and the parameter βp 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 *βp*. Note that if $\beta_\mathrm{p} \rightarrow \infty \quad \forall p$ (i.e., the second term of $\mathrm{J_m(U,Y)}$ is omitted), then a trivial solution is obtained by the minimization of the remaining cost function. The pair (U; Y) minimizes $\mathrm{J_m}$, under the above constraints only if:

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

and

$$y_p = \sum_{q=1}^{n} x_q\, u_{pq} \Big/ \sum_{q=1}^{n} u_{pq} \qquad \forall\, p. \qquad (3)$$

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

2.2 The Proposed Objective Function

The proposed solution of the problem resides in modifying the objective function in equ. (1) so that it minimizes the average dissimilarity between every pair of data points within the cluster.

Based on equ. (1) The objective function to be minimized is

$$J_m(U) = \sum_{p=1}^{k}(\sum_{i=1}^{n} u_{pi}(\frac{1}{\sum_{j\neq i}^{n} u_{pj}}\sum_{j\neq i}^{n} u_{pj}\, d(x_i,x_j)) + \beta_p\sum_{i=1}^{n}(u_{pi}\ln u_{pi} - u_{pi})) \qquad (4)$$

Equ. (4) is equivelent to the objective function of PCM except that $\mathrm{d}(x_i, v_p)$ is replace by:

$$\frac{1}{\sum_{j\neq i}^{n} u_{pj}}\sum_{j\neq i}^{n} u_{pj}\, d(x_i,x_j)$$

By setting the derivative to zero

$$\frac{\partial \mathrm{J_m}}{\partial u_{\mathrm{pi}}} = \frac{1}{\sum_{j\neq i}^{n} u_{pj}}\sum_{j\neq i}^{n} u_{pj}\, d(x_i,x_j) + \beta_p \ln(u_{pi}) = 0 \qquad (5)$$

Note that $\frac{\partial \mathrm{J_m}}{\partial u_{\mathrm{pi}}}(u_{pi}\ln(u_{pi}) - u_{pi}) = 1 + \ln(u_{pi}) - 1 = \ln(u_{pi})$

also the derivative of any sub-term of the first term of equ. (4) which does not contain u_{pi} is equal to zero.

The solution of equation (5) is

$$u_{pi} = e^{-E_{pi} / \beta_p} \qquad \forall\ p,i \qquad (6)$$

Where

$$E_{pi} = \frac{1}{\sum_{j \neq i}^{n} u_{pj}} \sum_{j \neq i}^{n} u_{pj}\, d(x_i, x_j) \qquad (7)$$

In section 2.5, equ. (6) and (7) are used as formulas for recalculating the membership functions. Initial Membership matrix M and β_p are computed using the proposed initialization procedure described in the next section. S, M and β_p are given as input to the possibilistic algorithm AVLINK in Fig. 4.

2.3 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

```
procedure initproc
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 represent initial clustering
    c : number of clusters
begin [initproc]
    1. Compute total similarity for each pattern
    for $j$=1, 2, ..., $n$   $Sum(x_i) = \sum_{j \neq i}^{n} S(x_i, x_j)$
    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
         Remove $x_j = \text{argmin}_{x_i \in C_k} \text{Sum}(x_i)$
         Mask $x_j$ and update totSum, Sum
         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
          set initial memberships in U to 0 for all outliers
          Return U
       else
          increment $k$
          update Sum, totSum, totCount //for unlabeled
          Go To Step 4
       endif
    end [initproc]
```

Fig. 1 The Proposed Initialization Procedure.

	S_1	S_2	S_3	S_4	sum
S_1	-	0.8	0.7	0.9	2.4
S_2	0.6	-	0.5	0.1	1.2
S_3	0.5	0.4	-	0.6	1.5
S_4	0.8	0.6	0.7	-	2.1
-	-	-	-	-	7.2

(a)

	S_1	S_2	S_3	S_4	sum
S_1	-	-	-	0.9	0.9
S_2	-	-	0.5	-	-
S_3	-	0.4	-	-	-
S_4	0.8	-	-	-	0.8
-	-	-	-	-	1.7

(b)

Fig. 2 Example for initialization step, (a) initial sum array, (b) sum after removing s2, s3.

potential outliers. Only steps 3-6 need to be repeated if the count of candidate outliers exceeds the expected percentage of outliers in the dataset by reducing the input threshold. For example: assuming α =0.75, in Fig.2(a) an asymmetric similarity matrix(computed using Dor so the diagonal are ones) of four subsequences s1..s4 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 the candidate 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 because (1.7/2) = 0.85 > 0.75).

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.

2.4 Estimating the Threshold α

The main drawbacks of the initialization procedure in [13] are that it needs several parameters; a user cannot easily specify a suitable value for them. This section describes 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. 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_p is to use the average E_{pi} for random sample of objects in the identified subsets and assuming membership value equals 0.3, as follows:

$$\beta_p = \frac{-\ln 0.3}{average\,E_{pi}} \qquad (8)$$

2.5 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. (6) and (7).

Finally, the algorithm terminates if no significant change in memberships which is decided by the input parameter ε .

2.5 Dissimilarity Measures

The non-gapped pair-wise homology alignment score $h(x_i, v_j)$ is a similarity score between two subsequences x_i and v_j [13] and it is defined as follows:

$$h(x_i, v_j) = \sum_{k=1}^{d} M(x_{ik}, v_{jk}) \qquad (9)$$

The corresponding dissimilarity h′(x_i,v_j) is:

```
Procedure EstimateAlphaBeta
input:
    Seq: is the whole input sequence (string of alphabetic)
    confd: very small real number between 0 and 1
output:
    S : Similarity matrix of size $n×n$
    $\alpha$ : threshold on the average similarity
    $\beta$ : parameters of the objective function in equ. (4)
begin [EstimateAlphaBeta]
    1. Compute the Similarity matrix S using Dor
    2. Select randomly very large number (thousands) of subsets
    of input subsequences.
    3. Compute 1000 bins histogram for the average
    similarities for each subset selected by step 2.
    4. Get the number of last bins bins from the right end of
    the histogram such that their sum = confd * totalcount.
    5. Compute $\alpha$ = max. value - bins * step (a value close to
    the upper 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
    7. Compute initial $\beta_p$ using equ. (8)
    End [EstimateAlphaBeta]
```

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

```
Input:
    S /*Similarity matrix of size $n×n$ */
    $\beta_p$ /*the parameters of (6) estimated in Fig. 3 */
    $\varepsilon$ /* the threshold controlling the convergence*/
    k /* the number of clusters to be found*/
    U /*initial membership matrix produced as in Fig. 1*/
Output:
    U /* matrix of possibilistic memberships*/
Begin [AVLINK]
    Repeat
        store Memberships $U$ in $U'$
        for each object $x_i$
          begin
            Compute $E_{pi}$ for $p$ = 1,2,3...$k$ using equ. (7)
            Compute $u_{pi}$= $e^{-E_{pi}/\beta_p}$ using euq. (6)
          end
          Update $\beta_p$ using equ. (8) for $p$=1,2,…$k$
        end
    Until $(\|U - U'\| < \varepsilon)$
    Output U
    End [AVLINK]
```

Fig. 4 Proposed Clustering Algorithm (AVLINK).

$$h'(x_i, v_j) = h(x_i, x_i) - h(x_i, v_j) \qquad (10)$$

Where d is the number of characters in the subsequences and is set to 8 in the preprocessing step of [13]. M(x_{ik}, x_{jk}) is a homology alignment score (similarity value) and can be obtained from a table lookup called mutation matrix [16]. A mutation matrix has 20 columns and 20 rows. M(a,b) is the value at the ith row and jth column of the mutation matrix where i, j correspond to the two alphabets a, b respectively. M(a,b) is integer value that represents the probability or a likelihood value that the amino acid a mutates to the amino acid b after a particular evolutionary time [10]. Each character in a subsequence corresponds to row/column in the mutation matrix. Also the mutation matrix is asymmetric which implies $h(x_i, x_j) \neq h(x_j, x_i)$ i.e. the similarity matrix using h is actually a complete one not an upper triangular one.

Another similarity measure between two subsequences is the ratio between the non-gapped pair-wise homology alignment scores of two input subsequences x_i and v_j to the maximum homology alignment score of the subsequence v_j [13] and is defined as follows:

$$DOR\ (x_i, v_j) = h(x_i, v_j) / h(v_j, v_j) \qquad (11)$$

The corresponding distance DDOR is defined as follows:

$$DDOR\ (x_i, v_j) = h'(x_i, v_j) / h(v_j, v_j) \qquad (12)$$

Where

DDor(x,x)=0

$0 \leq$ DDOR(x_i,v_j)<1.

DDOR(x_i,v_j)≠ DDOR(v_j, x_i).

3. Results and Discussion

The following is a list of the algorithms that are used in analyzing the performance of AVLINK:

(1) C-medoids(RFCMdd, FCMdd, HCMdd and RCMdd [13])

(2) Neural Network(MI) [14]

(3) Genetic algorithm(GAFR) [20]

RFCMdd is re-implemented to allow comparing execution time. In comparing with GAFR and MI the values reported in [13] are used. Finally, the proposed algorithm and RFCMdd are implemented using C# and run in windows 7, 64-bits environment having a machine configuration of core I3, 2.4 GHz, 1 Mbyte cache, and 4GB of RAM.

3.1 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 [6] 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_057850 represents 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.

3.2 *Quality Measures*

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

$$\beta = (1/c)\sum_{i=1}^{c}(1/n_i)\sum_{j=1}^{n} DOR(x_j, v_i) \qquad (13)$$

$$\gamma = \max_{i,j} \frac{1}{2}(DOR(v_j, v_i) + DOR(v_i, v_j)) \qquad (14)$$

The higher the value of β the better the quality of clustering. Also the lower the value of γ, the better the quality.

3.*3 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 AVLINK while the initial procedure of [13] is used in comparing with C-Medoids.

In Table 1, the values for the other algorithms are those reported in [13]. All the reported results in table 1 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 AVLINK is initialized using the proposed procedure.

Table 1 Performance of AVLINK compared to all above listed Algorithms on NP_057849.

Algorithm	Param.	B	γ	Param.	β	γ
AVLINK		0.801	0.799		0.873	0.789
RFCMdd		0.736	0.914		0.801	0.819
FCMdd		0.719	0.914		0.746	0.828
RCMdd	C=13	0.612	0.938	C=27	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
AVLINK		0.825	0.815		0.891	0.672
RFCMdd		0.801	0.821		0.836	0.681
FCMdd		0.746	0.837		0.767	0.701
RCMdd	C=26	0.632	0.836	C=36	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

Table 2 Execution Time in (ms) for different number of clusters compared to RFCMdd on NP_057849 and NP_057850.

Cluster Count	NP_057849		NP_057850	
	AVLINK	RFCMdd	AVLINK	RFCMdd
6	08122	05213	0971	0882
13	24223	13612	1533	1423
26	38312	24063	3677	3114
36	50802	32713	4122	3821
50	71507	44329	6231	5754

A 20 runs were executed to get a reliable average measure of the validation indices for AVLINK at various numbers of clusters. The value of the 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, it was generated using linear interpolation. The proposed algorithm scored higher values for β and lower values for γ and proved to be superior to other tested algorithms. The gain in β was higher than for γ. The results in table 2 represents the average execution time for AVLINK and RFCMdd In 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 2. It is clear that AVLINK is slightly slower than RFCMdd on NP_057849 and comparable to it for the smaller dataset NP_057850. Also the higher the cluster counts the higher the execution time for both algorithms.

Even though the computation of the average 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 AVLINK has a higher convergence rate than medoids-based algorithms.

By computing the similarity measure at the beginning and by maintaining the row sum and column sum of the similarity along with the cluster sum throughout the course, the runtime of the proposed algorithm becomes slightly higher than RFCMdd.

4. Conclusions

This paper presented a novel robust clustering algorithm along with an initialization procedure in which a threshold on the average similarity rather than a pre-specified number of clusters is specified. By applying the proposed algorithm in biological sequences analysis and comparing its results with the results obtained for other classical and state-of the-art clustering algorithms, the proposed clustering algorithm showed remarkable performance and proved to be competitive to other widely used algorithms for biological 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 than C-Medoids algorithms.
- The algorithm AVLINK is compared to GAFR and MI in terms of quality measured as β, γ and shows highly competitive results.
- By keeping the rows and columns basis of the membership matrix, AVLINK is slightly slower than RFCMdd.
- AVLINK needs less number of parameters than C-Medoids algorithms and the few parameters that are needed can be systematically computed or fine-tuned.

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]. S. Henikoff and J.G. Henikoff, "Amino Acid Substitution Matrices from Protein Blocks," Proc. Nat'l Academy of Sciences (PNAS '92), vol. 89, pp. 10915-10919, 1992.

[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]. R. Krishnapuram, J. M. Keller "A possibilistic approach to clustering," Fuzzy Systems, IEEE Transactions on 1(2) (1993) 98-110.

[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]. R. Krishnapuram, J. M. Keller, "The possibilistic

c-means algorithm: insights and recommendations," Fuzzy Systems, IEEE Transactions, 1996, pp. 385-393.

[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 Aelst S and Zha H "Machine learning and robust data mining," Computer Statistics and Data Analysis, vol. 52, pp.151-154, 2007.

[27]. Schynsa M, Haesbroeck G, Critchley F 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. Sharara M.A.Ismail, Biosoft: "αCORR: A novel algorithm for clustering gene expression data," Bioinformatics and Bioengineering,

[30]. 2007. BIBE 2007. Proceedings of the 7th IEEE International Conference, pp. 974-981, 2007.

[31]. 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.

[32]. Ling-Hong Hung, and Ram Samudrala "fast_protein_cluster: parallel and optimized clustering of large scale protein modeling data," Bioinformatics, 2014.

[33]. J. C. Dunn, "Well separated clusters and fuzzy partitions", Journal of Cybernetics, 4 (95-104), 1974.

[34]. 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.