JASEM ISSN 1119-8362

Full-text Available Online at
www.ajol.info and
www.bioline.org.br/ja

J. Appl. Sci. Environ. Manage. *December 2013*
Vol. 17 (4) 537-548

On an Improved Fuzzy C-Means Clustering Algorithm

***[1]FELIX UKPAI OGBAN; PRINCE OGHENEKARO ASAGBA(PH.D.); OLUMIDE OWOLABI(PH.D.)**

[1]Department of Mathematics, Statistics, & Computer Science, Faculty of Science
University of Calabar, Nigeria
Email: felix.ogban@gmail.com
[2]Department of Computer Science, Faculty of Physical & Information Technology, University of Port Harcourt, Nigeria
Email: pasagba@yahoo.com

[3]Computer Center, University of Abuja, Nigeria

Keywords: Fuzzy clusters, unsupervised learning, classification, similarity measures, Page classification.

ABSTRACT: A cluster *is a* gathering of similar objects which can exhibit dissimilarity to the objects of other clusters. Clustering algorithms may be classified as: Exclusive, Overlapping, Hierarchical, and Probabilistic; and several algorithms have been formulated for classification and found useful in different areas of application. The K-means, Fuzzy C-means, Hierarchical clustering, and Mixture of Gaussians are the most prominent of them. Our interest on this work is on the web search engines. In this paper, we examined the fuzzy c-means clustering algorithm in anticipation to improving upon its application area. On the Web, classification of page content is essential to focused crawling. Focused crawling supports the development of web directories, to topic-specific web link analysis, and to analysis of the topical structure of the Web. Web page classification can also help improve the quality of web search. Page classification is the process of assigning a page to one or more predefined category label. In all, the tendency for a web page to contain the qualities of two or more clusters could exist. Thus exclusive clustering would not be very useful in our case; so the need for overlapping clustering using Fuzzy C-means. *It is worthy of note that the Fuzzy C-mean being an optimization problem,* converges to a local minimum or a saddle point. The iteration in some cases becomes recurring. At such a point, one would assume the saddle point is reached and if the iteration is not terminated, the loop may continue to a stack-grab that may fault (increase running time, etc) the algorithm. In this work, we developed a modified fuzzy C-mean clustering algorithm with a sharp stopping condition which was tested on a demo data to ascertain its convergence and comparatively test its efficiency. Corel Q-pro optimizer was used on a timing macro. Our result(s) are quite interesting and challenging as they clearly show the presence of inter-lapping documents along the spectrum of two different clusters. © JASEM

http://dx.doi.org/10.4314/jasem.v17i4.11

Clustering can be considered the most important *unsupervised learning* problem; so, as every other problem of this kind, it deals with finding a *structure* in a collection of unlabeled data. A loose definition of clustering could be "the process of organizing objects into groups whose members are similar in some way". A *cluster* is therefore a collection of objects which are "similar" between them and are "dissimilar" to the objects belonging to other clusters. Figure 1 shows clustering as a kind of similarity measures

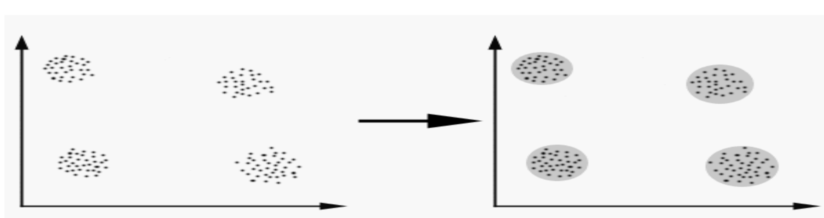


Fig. 1: Clustering as a kind of Similarity measures

Corresponding author Email: Email: felix.ogban@gmail.com

Source: (Tariq, 2002). In this case, we easily identify the 4 clusters into which the data can be divided; the similarity criterion is *distance*: two or more objects belong to the same cluster if they are "close" according to a given distance (in this case geometrical distance). This is called *distance-based clustering*. Another kind of clustering is *conceptual clustering*: two or more objects belong to the same cluster if this one defines a concept *common* to all that objects. In other words, objects are grouped according to their fit to descriptive concepts, not according to simple similarity measures. Page classification also known as web page classification is the process of assigning a page to one or more predefined category label. The field is often posed as a supervised learning problem. In all, the tendency for a web page to contain the qualities of two or more clusters could exist. Thus exclusive clustering would not be very useful in our case; so the need for overlapping clustering using Fuzzy C-means. In this work, a modified fuzzy C-mean clustering algorithm with a sharp stopping condition was introduced and tested on a demo data to ascertain its convergence and comparatively test its efficiency. Corel Q-pro optimizer was used on a timing macro.

The Goals Of Clustering: The goal of clustering is to determine the intrinsic grouping in a set of unlabeled data. But how to decide what constitutes a good clustering? It can be shown that there is no absolute "best" criterion which would be independent of the final aim of the clustering. Consequently, it is the user which must supply this criterion, in such a way that the result of the clustering will suit their needs. For instance, we could be interested in finding representatives for homogeneous groups (*data reduction*), in finding "natural clusters" and describe their unknown properties (*"natural" data types*), in finding useful and suitable groupings (*"useful" data classes*) or in finding unusual data objects (*outlier detection*).

Possible Applications

Clustering algorithms can be applied in many fields, such as:

Marketing: finding groups of customers with similar behavior given a large database of customer data containing their properties and past buying records;

Biology: classification of plants and animals given their features;

Libraries: book ordering;

Insurance: identifying groups of motor insurance policy holders with a high average claim cost; identifying frauds;

City-planning: identifying groups of houses according to their house type, value and geographical location;

Earthquake studies: clustering observed earthquake epicenters to identify dangerous zones;

WWW: document classification; clustering web log data to discover groups of similar access patterns.

Requirements

The fundamental requirements that a clustering algorithm should satisfy are: scalability; dealing with different types of attributes; discovering clusters with arbitrary shape; minimal requirements for domain knowledge to determine input parameters; ability to deal with noise and outliers; insensitivity to order of input records; high dimensionality; interpretability and usability.

Problems

There are a number of problems with clustering; some of them are: current clustering techniques do not address all the requirements adequately (and concurrently); dealing with large number of dimensions and large number of data items can be problematic because of time complexity; the effectiveness of the method depends on the definition of "distance" (for distance-based clustering); if an *obvious* distance measure doesn't exist we must "define" it, which is not always easy, especially in multi-dimensional spaces;

The result of the clustering algorithm (that in many cases can be arbitrary itself) can be interpreted in different ways.

Clustering Algorithms: Classification; Clustering algorithms may be classified as: Exclusive, Overlapping, Hierarchical, or Probabilistic.

Exclusive Clustering: In Exclusive Clustering, the first case data are grouped in an exclusive way, so that if a certain datum belongs to a definite cluster then it could not be included in another cluster. Figure 2 shows a Bi-dimensional plane in exclusive clustering, where the separation of points is achieved by a straight line on a bi-dimensional plane.

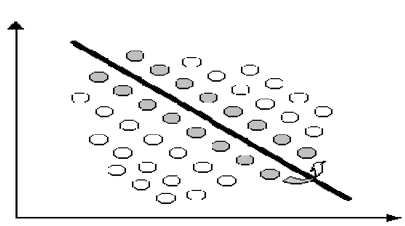


Fig. 2: A Bi-dimensional plane in exclusive clustering
Source: (Tariq, 2002)

Overlapping Clustering: On the contrary the second type, the overlapping clustering, uses fuzzy sets to cluster data, so that each point may belong to two or more clusters with different degrees of membership. In this case, data will be associated to an appropriate membership value.

Hierarchical Clustering: Instead, a hierarchical clustering algorithm is based on the union between the two nearest clusters. The beginning condition is realized by setting every datum as a cluster. After a few iterations it reaches the final clusters wanted.(Hans-Joachim and Hizir; 2007).

Probabilistic Clustering : Finally, the last kind of clustering uses a completely probabilistic approach. Here a probability distribution function is used to assign to each data of a cluster depending in its closeness.

Several algorithms have been formulated (for each of the above highlighted clustering classification) and found useful in different areas of application. Most common of them all are: K-means, Fuzzy C-means. Hierarchical clustering, Mixture of Gaussians

Each of these algorithms belongs to one of the clustering types listed above. So that, K-means is an *exclusive clustering* algorithm, Fuzzy C-means is an *overlapping clustering* algorithm, Hierarchical clustering is obvious and lastly Mixture of Gaussian is a *probabilistic clustering* algorithm (Tariq; 2002). In this work, we considered the Fuzzy C-means because recent clustering applications pose the characteristics of their data elements, exhibiting the tendencies of belonging to two or more clusters at the same context if well considered.

Distance Measure: An important component of a clustering algorithm is the distance measure between data points (Osmar; 2006). If the components of the data instance vectors are all in the same physical units then it is possible that the simple Euclidean distance metric is sufficient to successfully group similar data instances. However, even in this case the Euclidean distance can sometimes be misleading. Figure 3 shows the scaling of the width and height measurements of an object. Despite both measurements being taken in the same physical units, an informed decision has to be made as to the relative scaling. Different scaling can lead to different clusters.

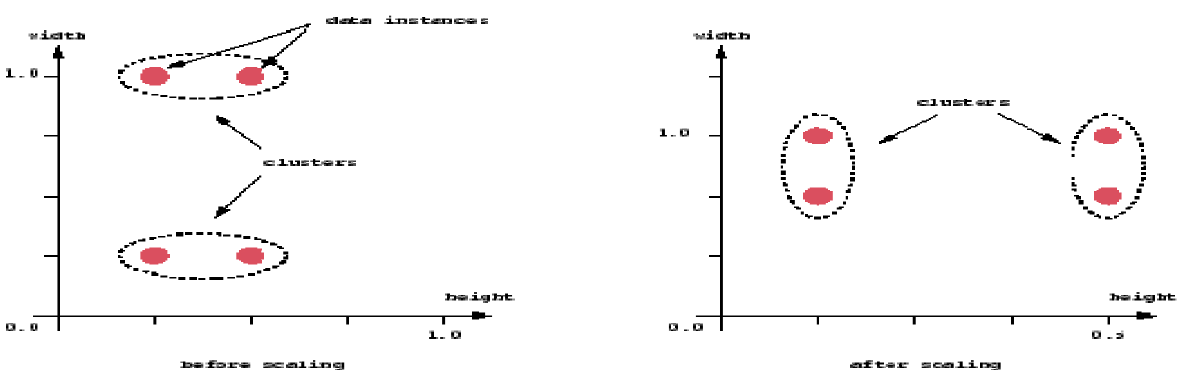


Fig. 3: Scaling of the width and height measurements of an object

Notice however that this is not only a graphic issue: the problem arises from the mathematical formula used to combine the distances between the single components of the data feature vectors into a unique distance measure that can be used for clustering purposes: different formulas leads to different clustering(s). Again, domain knowledge must be used to guide the formulation of a suitable distance measure for each particular application.

Minkowski Metric : For higher dimensional data, a popular measure is the Minkowski metric,

$$d_p(x_i, x_j) = \left(\sum_{k=1}^{d} |x_{i,k} - x_{j,k}|^p \right)^{\frac{1}{p}} \quad (1)$$

where d is the dimensionality of the data. The *Euclidean* distance is a special case where p=2, while *Manhattan* metric has p=1. However, there are no general theoretical guidelines for selecting a measure for any given application. It is often the case that the components of the data feature vectors are not immediately comparable. It can be that the components are not continuous variables, like length, but nominal categories, such as the days of the week. In these cases again, domain knowledge must be used to formulate an appropriate measure.

The Algorithm: Fuzzy c-means (FCM) is a method of clustering which allows one piece of data to elong to two or more clusters. This method (developed by Dunn in 1973 and improved by Bezdek in 1981) is frequently used in pattern recognition. It is based on minimization of the following objective function:

$$J_m = \sum_{i=1}^{N}\sum_{j=1}^{C} u_{ij}^m \|x_i - c_j\|^2, \quad 1 \le m < \infty \qquad (2)$$

where m is any real number greater than 1, u_{ij} is the degree of membership of x_i in the cluster j, x_i is the ith of d-dimensional measured data, c_j is the d-dimension center of the cluster, and $\|*\|$ is any norm expressing the similarity between any measured data and the center.

Fuzzy partitioning is carried out through an iterative optimization of the objective function shown above, with the update of membership u_{ij} and the cluster centers c_j by

:

$$u_{ij} = \frac{1}{\sum_{k=1}^{C}\left(\frac{\|x_i - c_j\|}{\|x_i - c_k\|}\right)^{\frac{2}{m-1}}}, \qquad (3)$$

$$c_j = \frac{\sum_{i=1}^{N} u_{ij}^m \cdot x_i}{\sum_{i=1}^{N} u_{ij}^m} \qquad (4)$$

This iteration will stop when

$$\max_{ij}\left\{\left|u_{ij}^{(k+1)} - u_{ij}^{(k)}\right|\right\} < \varepsilon, \qquad (5)$$

where ε is a termination criterion between 0 and 1, whereas k are the iteration steps.

This procedure converges to a local minimum or a saddle point of J_m. The Fuzzy c-means (FCM) clustering algorithm is shown in Figure 4.

1. *Initialize $U=[u_{ij}]$ matrix, $U^{(0)}$*
2. *At k-step: calculate the centers vectors $C^{(k)}=[c_j]$ with $U^{(k)}$*

$$c_j = \frac{\sum_{i=1}^{N} u_{ij}^m \cdot x_i}{\sum_{i=1}^{N} u_{ij}^m}$$

3. *Update $U^{(k)}$, $U^{(k+1)}$*

$$u_{ij} = \frac{1}{\sum_{k=1}^{C}\left(\frac{\|x_i - c_j\|}{\|x_i - c_k\|}\right)^{\frac{2}{m-1}}}$$

4. *If $\| U^{(k+1)} - U^{(k)}\| < \varepsilon$ then STOP; otherwise return to step 2.*

Fig.4: The Fuzzy c-means (FCM) clustering algorithm

Fuzzy Representation Of Data Sets: Data are bound to each cluster by means of a Membership Function, which represents the fuzzy behavior of this algorithm. To do that, we simply have to build an appropriate matrix named U whose factors are numbers between 0 and 1, and represent the degree of membership between data and centers of clusters. *For a better understanding, we may consider this* *simple mono-dimensional example. Given a certain data set, suppose to represent it as distributed on an axis.* Figure 4 shows (Fig. 4) axis representation of data sets.

Fig. 5: Axis representation of data sets

Looking at the picture, we may identify two clusters in proximity of the two data concentrations. We will refer to them using 'A' and 'B'as in Figure 6. In the first approach shown in this work - the k-means algorithm - we associated each datum to a specific centroid, with membership function described in Figure 6 - Centroid representation of the data sets .

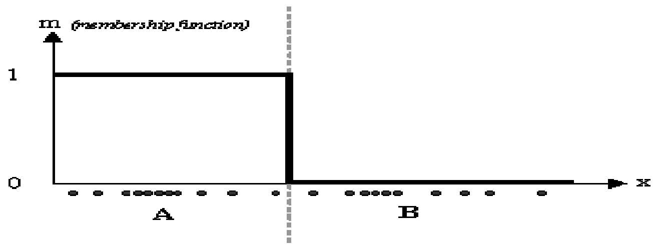


Fig. 6: Centroid representation of the data sets

In the FCM approach, instead, the same given datum does not belong exclusively to a well defined cluster, but it can be placed in a middle way. In this case, the membership function follows a smoother line to indicate that every datum may belong to several clusters with different values of the membership coefficient. Figure 7 shows the fuzzy c-means approach of centroid representation .

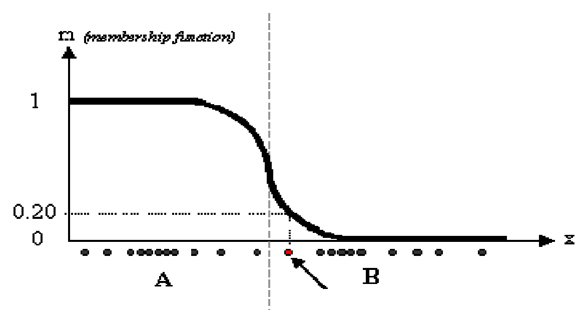


Fig. 7: The fuzzy c-means approach of centroid representation

In fig. 7, the datum shown as a red marked spot (pointed to by an arrow) belongs more to the B cluster rather than the A cluster. The value 0.2 of 'm' indicates the degree of membership to A for such datum. Now, instead of using a graphical representation, we introduce a matrix U whose factors are the ones taken from the membership functions as described in Figure 8.

$$U_{NxC} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ .. & .. \\ 0 & 1 \end{bmatrix} \qquad U_{NxC} = \begin{bmatrix} 0.8 & 0.2 \\ 0.3 & 0.7 \\ 0.6 & 0.4 \\ .. & .. \\ 0.9 & 0.1 \end{bmatrix}$$

(a) (b)

Fig: 8: A matrix U whose factors are taken from the membership functions

The number of rows and columns depends on how many data and clusters we are considering. More exactly we have C = 2 columns (C = 2 clusters) and N rows, where C is the total number of clusters and N is the total number of data. The generic element is so indicated: u_{ij}. In Figure 8, we have considered the k-means (a) and FCM (b) cases. We can notice that in the first case (a) the coefficients are always unitary. It is so to indicate the fact that each datum can belong only to one cluster. Other properties of k-means, is shown in Figure 9.

- $u_{ij} \in [0,1] \quad \forall i, j$
- $\sum_{j=1}^{C} u_{ik} = 1 \quad \forall i$
- $0 < \sum_{i=1}^{N} u_{ij} < N \quad \forall N$

***Fig. 9**: Other* properties of k-means

The Modification/Improvement Of The Fuzzy C-Means Clustering Algorithm: It must be noted *the Fuzzy C-mean being an optimization problem,* converges to a local minimum or a saddle point of J_m as spelled out in Equation 2 above. The iteration in some cases becomes recurring. At such a point, one would assume the saddle point is reached and if the iteration is not terminated, the loop may continues to a stack-grab that may fault (increase running time, etc) the algorithm. Note also that for every center vector [Cj], we do not only consider the center membership function of the datum but also its associative properties that should make the same datum in consideration acceptable in another cluster (overlap). With these and other problem not discussed here. A modified Fuzzy C-mean algorithm is shown in Figure 10.

1. *Initialize U=[u_{ij}] matrix, $U^{(0)}$*
2. *At k-step: calculate the centers vectors $C^{(k)}$=[c_j] with $U^{(k)}$*

$$c_j = \frac{\sum_{i=1}^{N} u_{ij}^{m} \cdot x_i}{\sum_{i=1}^{N} u_{ij}^{m}}$$

3. *Update $U^{(k)}$, $U^{(k+1)}$*

$$u_{ij} = \frac{1}{\sum_{k=1}^{C} \left(\frac{\| x_i - c_j \|}{\| x_i - c_k \|} \right)^{\frac{2}{m-1}}}$$

4. *If $\| U^{(k+1)} - U^{(k)} \| < \varepsilon$ then STOP;*
5. *else If $\| U^{(k+1)} - U^{(k)} \|_{new} = \| U^{(k+1)} - U^{(k)} \|_{old}$ then STOP else return to step 2.*

Fig. 10: A modified Fuzzy C-mean algorithm

Increasing the STOP conditions would help reduce memory consumption in the execution frame of the optimization problem.

A case Study Result and discussion.: Here, we considered the simple case of a mono-dimensional application of the FCM. Forty by ten array data as shown in Table 1 below. Three clusters were used to initialize the algorithm and to compute the U matrix

.

Table 1: A Mono-Dimensional Collection of Forty by Ten Array Data

1	32.97296	33.78795	33.07803	34.07537	34.75031	32.11693	33.22848	34.43884	33.83738	32.69207
2	33.96386	33.49916	33.04058	34.60142	32.31941	33.96008	33.65453	32.13929	33.84962	33.86212
3	33.27152	34.48831	33.41839	32.00967	33.80071	32.86538	34.11624	33.09643	32.71353	33.16603
4	33.91233	34.18517	32.31239	33.15703	34.59071	34.70568	34.68988	33.64537	34.19484	32.70432
5	33.50952	32.64731	34.49717	34.38035	32.58036	34.16499	33.68805	34.41882	33.68195	34.54783
6	33.00637	33.65958	33.80495	32.85256	33.03735	32.86458	32.7068	32.01511	34.09383	33.78251
7	34.54749	34.59438	33.6421	33.46564	33.59221	34.28203	33.84567	34.7824	34.18129	32.08487
8	34.64431	33.999	33.66636	33.19013	33.73534	34.77561	33.80945	34.48363	33.45283	33.46029
9	32.1483	33.85409	33.16394	34.33249	32.0625	32.85894	33.89253	33.42307	32.00949	32.79355
10	32.56908	34.08601	32.00809	33.97171	34.18223	33.14778	34.06555	32.7857	32.99277	32.17334
11	33.53474	32.93372	32.82803	32.60112	34.85034	32.40954	32.58042	32.55856	32.3268	33.06373
12	33.28613	33.87954	34.69329	34.63844	32.86184	33.89364	32.96245	34.29576	32.34776	32.77397
13	33.65951	33.43536	33.18406	34.41067	32.87307	33.5014	32.08486	33.34898	34.06129	34.17026
14	33.0233	34.60829	33.02689	32.55454	34.62432	33.36989	32.1489	32.31745	33.04703	32.40374
15	32.05233	32.35248	33.6414	33.7746	34.45659	33.7651	32.88112	34.33268	32.7706	32.19736
16	34.29613	33.96899	34.54854	32.05482	33.45477	33.04382	32.5758	33.56375	32.76699	34.85529
17	32.0605	32.77672	33.5286	34.3666	33.81987	32.2003	34.50204	32.59049	34.65484	32.29228
18	32.32145	33.61892	34.76016	32.5682	34.19747	32.20205	32.37206	34.62855	34.53784	33.12991
19	33.40782	32.22161	34.79025	32.87887	33.07182	34.73985	34.19144	34.06873	34.40285	33.04145
20	34.54701	32.28655	34.0565	33.83879	34.2818	32.46039	33.26662	32.28309	33.27674	32.02704
21	34.72519	34.65895	33.02943	33.17186	32.11079	32.7641	34.71571	33.46454	32.60655	33.39804
22	32.02967	32.81181	34.30236	34.03204	33.52733	34.72273	32.91775	34.66782	32.69412	33.7677
23	34.31911	33.76428	34.67156	32.09482	34.28079	34.11194	32.04546	34.28645	32.90106	33.4313
24	33.55697	34.56007	33.26597	34.13749	33.65167	34.11824	32.11721	32.36122	32.10935	32.99197
25	32.80313	33.63632	32.65289	33.71991	32.67939	33.63679	32.56294	32.55312	32.70141	33.41859
26	34.79298	34.57014	32.28917	33.36858	33.95753	33.10856	34.63351	34.15661	33.40306	32.09283
27	32.09964	33.09115	33.81552	32.48291	34.45099	34.59562	32.63445	32.92449	33.70428	33.96793
28	34.72298	34.48836	32.33593	34.62298	34.60524	33.54802	33.52339	32.69709	34.57655	33.39925
29	32.58929	32.31046	33.63343	32.9276	32.68875	34.19878	34.38683	33.35236	33.43921	32.45056
30	33.56502	33.43993	33.43106	32.54614	34.37222	32.68002	33.58226	32.90943	34.74113	32.2553
31	33.95145	33.10304	34.479	33.32884	32.3377	34.2761	32.31347	33.87546	32.21338	32.62849
32	33.25879	32.07693	34.3814	32.97501	34.59179	33.02479	34.64224	33.35917	34.78824	34.5066
33	32.09965	33.33743	32.00494	32.63502	33.32741	33.63716	34.81987	32.57341	32.40354	34.39045
34	32.19698	34.41841	32.15092	33.98183	34.85793	34.05626	32.43139	32.9717	32.66223	32.50512
35	34.40447	34.27864	34.1603	34.00547	33.37514	32.82413	34.10963	33.92888	33.4428	33.37078
36	33.10042	34.70814	32.2737	32.13653	33.80763	33.7185	33.59849	32.79778	32.69246	32.85419
37	32.6211	34.01806	34.67419	33.61038	33.26409	33.51044	33.59493	34.67088	33.57289	34.18296
38	34.15606	34.14808	32.04185	32.8152	33.99701	32.81871	32.21059	32.32754	32.05499	33.91762
39	33.66072	33.56452	32.42005	32.48916	34.69157	33.77032	32.20812	32.64933	34.61506	34.45302
40	34.41031	33.99937	33.64607	34.21778	32.63836	32.45162	32.56493	33.71947	34.67022	34.06824

Calculating the *centers vectors* $C^{(k)}=[c_j]$*, which is given by equation 4 above, will produce our table 2 shown below. Table 2 could be represented in a matrix format to aid the computation of the different Uij* iteratively. Equation 3 is an expression of U_{ij}. The number of iterations with *respect to the stopping rule of equation 5, could cause the function to converge.*

Table 2: Calculated *centers vectors* $C^{(k)}=[c_j]$ *for the forty by ten data of table 1*

0.029521	0.031021	0.029425	0.028693	0.029879	0.029496	0.029578	0.030058	0.029839	0.029023
0.030665	0.03109	0.031244	0.031086	0.028933	0.029216	0.030978	0.029377	0.029757	0.029496
0.031137	0.031056	0.030373	0.029304	0.030766	0.031119	0.029351	0.029112	0.030985	0.030586
0.030573	0.029628	0.029641	0.029181	0.029468	0.02945	0.029178	0.031112	0.029156	0.029487
0.029142	0.029155	0.030212	0.029558	0.029943	0.028955	0.02932	0.029212	0.029678	0.030956
0.029139	0.029891	0.028793	0.028826	0.029453	0.030405	0.03054	0.030209	0.029935	0.031233
0.031089	0.031102	0.031029	0.031042	0.030719	0.030291	0.028777	0.03063	0.030043	0.030197
0.029459	0.030795	0.029128	0.029519	0.029107	0.030841	0.029655	0.028976	0.029932	0.028815
0.028751	0.029335	0.029726	0.030865	0.030183	0.029832	0.031114	0.02884	0.029673	0.028906
0.030393	0.030332	0.030208	0.029538	0.028824	0.029281	0.028786	0.028866	0.028709	0.030137
0.03112	0.030839	0.029163	0.030271	0.03067	0.030286	0.028686	0.028673	0.030625	0.029381
0.03076	0.031101	0.030121	0.029821	0.02902	0.029635	0.030888	0.029114	0.030274	0.028786
0.029118	0.029268	0.031013	0.028931	0.028691	0.03064	0.03014	0.030428	0.030595	0.031201
0.03125	0.028695	0.029206	0.028682	0.028697	0.029523	0.029848	0.030679	0.02988	0.030358
0.031242	0.029258	0.029547	0.029821	0.03057	0.028837	0.029651	0.02963	0.030639	0.029383
0.030282	0.03115	0.030921	0.031046	0.029907	0.029637	0.029014	0.030598	0.02998	0.029532
0.028843	0.029485	0.029406	0.030516	0.030874	0.029163	0.02914	0.030769	0.030416	0.028805
0.02943	0.028904	0.029985	0.030467	0.030196	0.029727	0.030919	0.029059	0.029856	0.028826
0.030666	0.029648	0.029107	0.030035	0.030058	0.029952	0.030302	0.030367	0.029463	0.030203
0.03003	0.029874	0.028676	0.029285	0.029657	0.029746	0.029014	0.030458	0.030181	0.030248
0.029714	0.029816	0.029116	0.029308	0.028786	0.029666	0.029691	0.029554	0.029494	0.030432
0.029504	0.028846	0.030684	0.030111	0.02886	0.030676	0.030315	0.029988	0.031186	0.029794
0.028989	0.030766	0.02921	0.029378	0.028831	0.03037	0.02981	0.029186	0.030052	0.031171
0.030083	0.029884	0.029325	0.029877	0.030806	0.029384	0.030742	0.030374	0.03116	0.029891
0.029253	0.030111	0.02897	0.030604	0.029873	0.031162	0.029171	0.028792	0.030912	0.029987
0.029218	0.029702	0.030145	0.030651	0.02936	0.029694	0.031176	0.029447	0.03096	0.02972
0.028925	0.029941	0.029915	0.028777	0.029772	0.031106	0.02907	0.030012	0.028961	0.029029
0.030022	0.029052	0.03093	0.030806	0.030388	0.029943	0.030916	0.029507	0.030826	0.029813
0.030779	0.030213	0.028732	0.02963	0.03035	0.02933	0.029734	0.030066	0.028817	0.029173
0.028696	0.030641	0.030794	0.028731	0.031206	0.031	0.031176	0.031147	0.029174	0.030527
0.028702	0.030232	0.029847	0.031005	0.029782	0.030683	0.02918	0.029575	0.030085	0.030302
0.029089	0.029886	0.031221	0.030897	0.0305	0.030914	0.02985	0.031002	0.028856	0.029221
0.030847	0.029013	0.030239	0.02919	0.030628	0.030921	0.030541	0.028699	0.030481	0.030351
0.030468	0.031198	0.030616	0.030063	0.030065	0.030485	0.030498	0.029368	0.03022	0.029194
0.031075	0.030145	0.0295	0.030991	0.030528	0.030278	0.029605	0.029808	0.030447	0.029515
0.031243	0.029991	0.030587	0.028727	0.030665	0.030877	0.030397	0.030109	0.030623	0.030449
0.029415	0.030968	0.030277	0.029091	0.029456	0.029741	0.030147	0.030623	0.030376	0.03035
0.030078	0.029875	0.031134	0.029131	0.030018	0.029696	0.03084	0.028988	0.029784	0.029427
0.028854	0.028837	0.03119	0.030997	0.029945	0.030272	0.030052	0.03073	0.028889	0.030028
0.029431	0.028943	0.030831	0.028977	0.028788	0.029688	0.030169	0.029153	0.029858	0.028844

Figure 11 shows the indicator of the fuzziness of data elements. Figure 12 shows the better indicator at higher iteration - 8 steps, and Figure 13 shows better indicator at higher iteration-37 steps. Figures 11, 12 and 13 (taken from our interactive test) show the membership value for each datum and for each cluster. The color of the data is that of the nearest cluster according to the membership function.

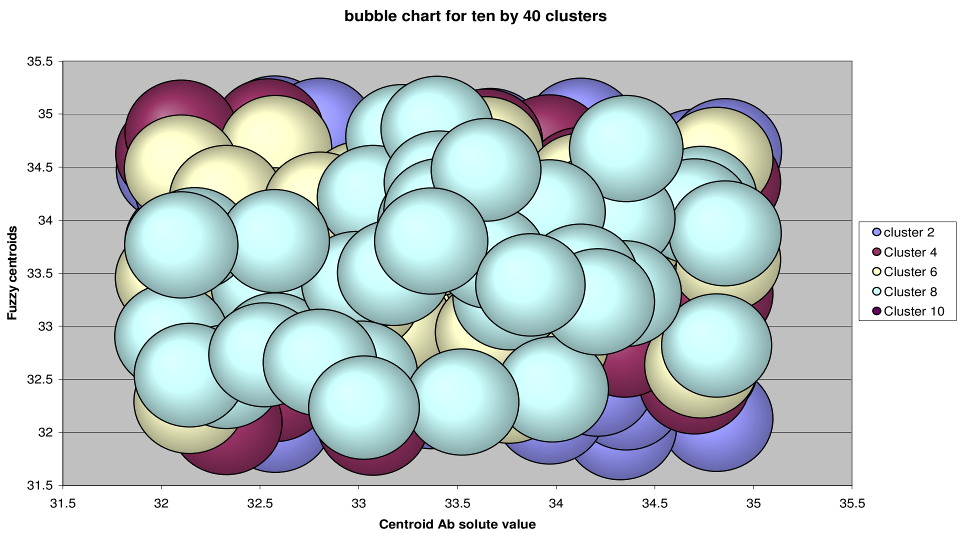


Fig. 11: Indicator of the fuzziness of data elements

In the simulation shown in Fig. 11 we have used a fuzziness coefficient m = 2 and we have also imposed to terminate the algorithm when

$$\max_{ij}\left\{\left|u_{ij}^{(k+1)} - u_{ij}^{(k)}\right|\right\} < 0.3$$

The picture shows the initial condition where the fuzzy distribution depends on the particular position of the clusters. No step is performed yet so that clusters are not identified very well. Now we can run the algorithm until the stop condition is verified. The fig. 12 shows the final condition reached at the 8th step with m=2 and ε = 0.3 using the smoothened values of table 2: Note that series 14 and 16 intersect at a point, indicating the line of fuzziness. However, series 20 did not at all intersect with that of 14 and 16, showing that no data element in series 20 has any link with that of 14 and 16.

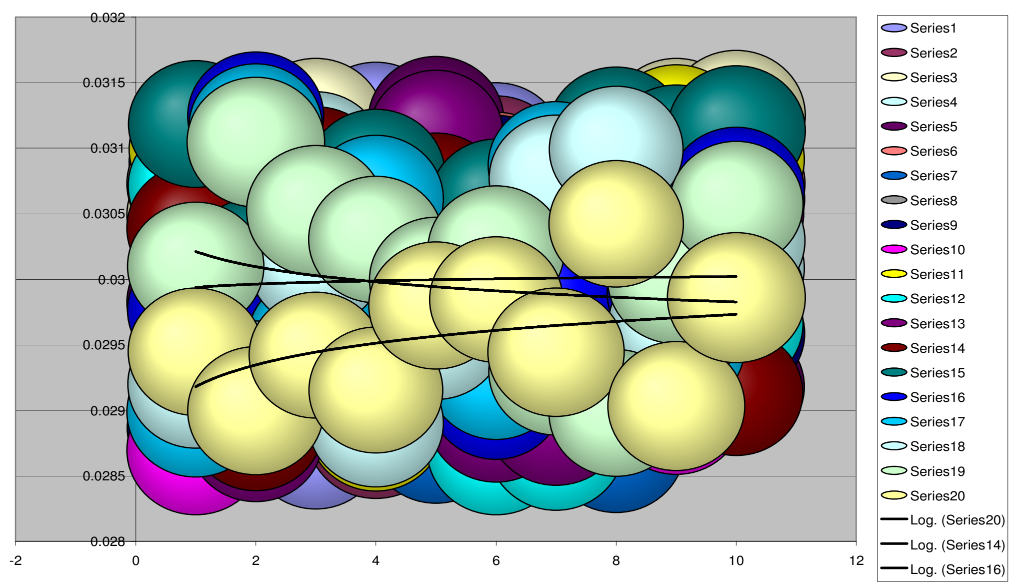
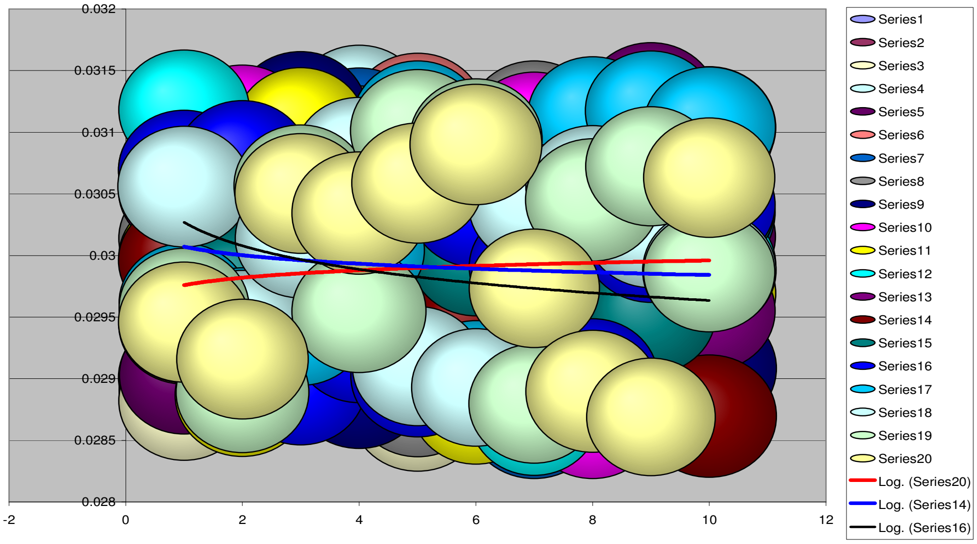


Fig.12: Better indicator at higher iteration -8 steps

Is it possible to do better? Certainly, we could use a higher accuracy but we would have also to pay for a bigger computational effort. In Figure 13 we can see a better result having used the same initial conditions and ε = 0.01, but we needed 37 steps of iterations. Note that series 14, 16 and 20 now intersect at a point, indicating the line of fuzziness for the three selected series. Therefore, as the iteration increases and the stopping rule not reached, series 20's center value converges towards a value in the cluster of series 14 and 16. Showing that, the computation of the center vector value causes the mining of new data elements in series 20



.

Fig. 13 Better indicator at higher iteration -37 steps

It is also important to notice that different initializations cause different evolutions of the algorithm. In fact it could converge to the same result but probably with a different number of iteration steps or alternatively converges to a new result and with a different number of iteration as shown in Figure 14, where series 14, 16 and 20 took new different trend at initial conditions of m = 2, and ε = 0.01 for -77 iterations

.

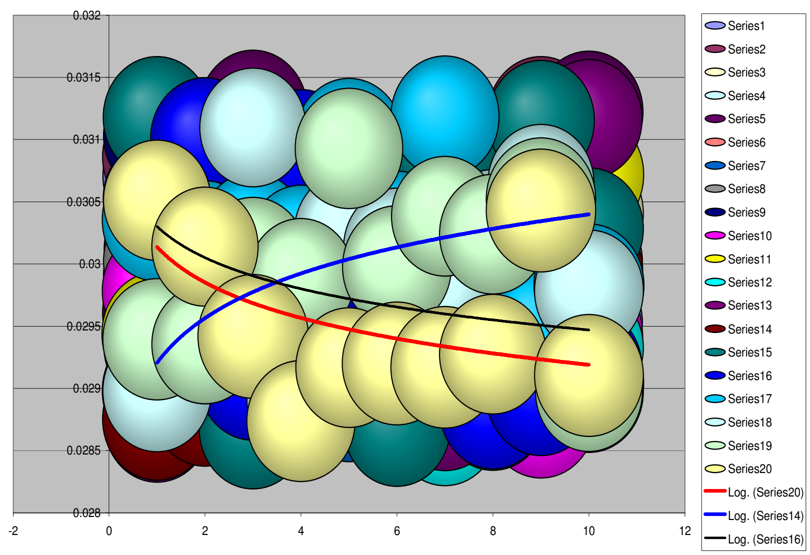


Fig. 14: Yet a Better indicator at higher iteration -77 steps

Conclusion: In this work, we developed a modified fuzzy C-mean clustering algorithm with a sharp stopping condition which was tested on a demo data to ascertain its convergence and comparatively test its efficiency

.

Corel Q-pro optimizer was used on a timing macro and micro soft excel graphs used to present the result. Our result(s) are quite interesting and challenging as they clearly show the presence of inter-lapping documents along the spectrum of two different clusters (inter and intra-cluster), indicating the usefulness of the fuzzy algorithm. As shown in Figures 11, 12, 13 and 14, the adjustment elasticity of the different constraints surrounding the convergence of the objective function can generate different results thus exposing the fuzziness of the data within and without the cluster. Out of forty trends of data links, only clusters 6 and 8 are fuzzy enough to link others. It is worthy of note that the Fuzzy C-mean being an optimization problem, converges to a local minimum or a saddle point. The iteration in some cases becomes recurring. At such a point, one would assume the saddle point is reached and if the iteration is not terminated, the loop may continue to a stack-grab that may fault (increase running time, etc) the algorithm.

REFERENCES

Abiteboul, S., Preda, M., and Cobena, G. (2003), Adaptive on-line page importance computation, *In*: Proceedings of the twelfth international conference on World Wide Web (Budapest, Hungary: ACM Press): 280–290.

Brin, S. and Page, L. (1998), The anatomy of a large-scale Hypertextual Web search engine. Computer Networks and ISDN Systems, 30(1-7):107–117.

Cho, J. and Garcia-Molina, H. (2003), Estimating frequency of change, ACM Transactions on Internet Technology, 3(3): 7 – 35

.

Ipeirotis, P., Ntoulas, A., Cho, J., Gravano, L. (2005), Modeling and managing content changes in text databases. In Proceedings of the 21st IEEE International Conference on Data Engineering, pages 606-617, Tokyo.

Lawrence, S. and Giles, C. L. (2000), Accessibility of information on the web, Intelligence, 11(1): 32–39.

Marc-Najork and Wiener J. L. (2001), Breadth-first crawling yields high-quality pages, In Proceedings of the Tenth Conference on World Wide Web, Elsevier Science, Hong Kong: 114–118.

Dunn, J. C. (1973), A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters, *Journal of Cybernetics* 3: 32-57.

Bezdek, J. C. (1981), Pattern Recognition with Fuzzy Objective Function Algorithms", Plenum Press, New York.

Tariq, R. (2002), Clustering: http://www.cs.bris.ac.uk/home/tr1690/documentation/fuzzy_clustering_initial_report/node11.html.

Hans-Joachim, M. A. and Hizir, S. (2007), Nonhierarchical Clustering: http://www.quantlet.com/mdstat/scripts/xag/html/xaghtmlframe149.html.

Osmar R. Z, and Zhanhuai L. (2006), Advanced Data Mining and Applications, published by Springer Verlag, Lecture Notes in Artificial Intelligence Volume 4093, .