Hungarian Association of Agricultural Informatics
European Federation for Information Technology in Agriculture,
Food and the Environment

Journal of Agricultural Informatics. 2015 Vol. 6, No. 3
journal.magisz.org

Comparison of K-Means and Fuzzy C-Means Algorithms on Different Cluster Structures

Zeynel Cebeci[1], Figen Yildiz[2]

INFO

Received: 24 June 2015
Accepted: 21 Aug 2015
Available on-line: 12 Oct 2015
Responsible Editor: M. Herdon

Keywords:
cluster analysis, fuzzy c-means, k-means, soft clustering, hard clustering.

ABSTRACT

In this paper the K-means (KM) and the Fuzzy C-means (FCM) algorithms were compared for their computing performance and clustering accuracy on different shaped cluster structures which are regularly and irregularly scattered in two dimensional space. While the accuracy of the KM with single pass was lower than those of the FCM, the KM with multiple starts showed nearly the same clustering accuracy with the FCM. Moreover the KM with multiple starts was extremely superior to the FCM in computing time in all datasets analyzed. Therefore, when well separated cluster structures spreading with regular patterns do exist in datasets the KM with multiple starts was recommended for cluster analysis because of its comparable accuracy and runtime performances.

1. Introduction

In recent years agricultural and environmental data have been increased in exponential rates by the widely use of automated data collection tools and systems. The yield data from precision agriculture applications have become one of the recent contributors in this increase. A huge amount of data collected by weather forecasting, remote sensing and geographic information systems have already been in use for a long time. In addition the progressive and intensive use of sensor networks and computers in the cultivated areas, barns and poultry houses have played a significant role in the increase of agricultural data. Because of this enormous growth, data mining (DM) techniques will be helpful to discover useful or meaningful information in agricultural big data. However, DM is a relatively novel field in agriculture, food, environment and other related areas (Ramesh &Vardhan 2013). Similar to the other areas such as pattern recognition, image segmentation, bio-informatics, web mining and consumer market research, Cluster Analysis (CA) as one of the most popular among many DM techniques could be used in agricultural data analysis. For instance, it is believed that DM and CA should be a part of agriculture because they can improve the accuracy of decision systems (Tiwari & Misra 2011).

As an umbrella term CA is defined as the collection of unsupervised classification techniques for grouping objects or segmenting datasets into subsets of data called as clusters. By using an appropriate clustering algorithm, a cluster is formed with objects which are more similar to each other when compared to others in different clusters. In other words, cluster analysis assigns similar objects into the same cluster which share common characteristics based on their features. Although there are some different ways to categorize them, the clustering algorithms can be generally grouped in 3 categories as hierarchical, non-hierarchical (flat) and mixture techniques. Although hundreds of algorithms do exist, in practice the use of many of these algorithms has been limited due to their complexity, efficiency and availability in presently used statistical software. The choice of a good algorithm to run on a certain dataset depends on many criteria such as data size, data structure, and the goals of CA (Velmurugan 2012; Bora & Gupta 2014). As reported in many studies (e.g. Dong *et al.* 2011; Kuar & Kuar 2013), the non-hierarchical partitioning algorithms, i.e. the algorithms belonging to K-means (KM) family give good clustering results in shorter times compared to the hierarchical algorithms on large datasets.

[1] Zeynel Cebeci
Div. of Biometry & Genetics, Faculty of Agriculture, Çukurova University, 01330 Adana - Turkey
zcebeci@cukurova.edu.tr, cebeciz@gmail.com

[2] Figen Yildiz
Div. of Biometry & Genetics, Faculty of Agriculture, Çukurova University, 01330 Adana - Turkey
yildizf@cukurova.edu.tr

doi: 10.17700/jai.2015.6.3.196

Therefore, since introduced by MacQueen (1967) KM and its successor derivatives have been the most popular algorithms in exploratory data analysis and DM applications over a half of century.

K-means (or alternatively Hard C-means after introduction of soft Fuzzy C-means clustering) is a well-known clustering algorithm that partitions a given dataset into c (or k) clusters. It needs a parameter c representing the number of clusters which should be known or determined as a fixed apriori value before going to cluster analysis. KM is reported fast, robust and simple to implement. As reported in many studies it gives comparatively good results if clusters in datasets are distinct or well separated. It was also examined that KM is relatively efficient in computational time complexity with its cost of $O(tcnp)$ in Lloyd algorithm (where *t*: number of iterations, *c*: number of clusters, *n*: number of objects, *p*: number of dimensions or number of features). Despite its above mentioned advantages, KM has several disadvantages too regarding the form and scattering of clusters in datasets. First of all, KM may not be successful to find overlapping clusters, and it is not invariant to non-linear transformations of data. For that reason, representations of a certain dataset with Cartesian coordinates and polar coordinates may give different clustering results. KM also fails to cluster noisy data and non-linear datasets.

In order to overcome some of the problems faced with KM, Bezdek (1981) introduced Fuzzy C-means (FCM) which is based on Dunn's study (Dunn 1973) as an extension of KM. As reviewed by Suganya & Shanthi (2012) and Ali *et al*. (2008), a dozen of the algorithms have been developed in order to improve the efficiency and accuracy of FCM. However, the basic FCM algorithm has frequently been used in a wide area of applications from engineering to economics. FCM is a soft algorithm clustering fuzzy data in which an object is not only a member of a cluster but member of many clusters in varying degree of membership as well. In this way, objects located on boundaries of clusters are not forced to fully belong to a certain cluster, but rather they can be member of many clusters with a partial membership degree between 0 and 1. In spite of its relatively higher cost with $O(tc^2np)$, when compared to KM, FCM has also been used in many clustering applications because of its above mentioned advantages in agriculture and forestry area (di Martino *et al*. 2007; 2009).

Although FCM is believed to be more efficient to analyze fuzzy data, it does not have a constant superiority in all cases of data structures according to the research findings. However, the recent studies generally have focused on comparison of KM and FCM by using some well-known test datasets such as Iris and Wine in R environment (Jipkate & Gohokar 2012; Panda *et al.* 2012; Ghosh & Dubey 2013; Bora & Gupta 2014). Thus, it would be helpful to examine these hard-and-soft C-means partitioning algorithms for the data structures following different patterns and shapes of clusters. For that reason, in this paper we compared the efficiency of KM and FCM algorithms on synthetically generated datasets consisting of different shaped clusters scattering with regular and non-regular patterns in two dimensional space.

2. K-means and Fuzzy C-means algorithms

Let $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n\}$ be a given dataset to be analyzed, and $\boldsymbol{V} = \{\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_c\}$ be the set of centers of clusters in $\boldsymbol{X}$ dataset in p dimensional space ($\mathbb{R}^p$). Where n is the number of objects, p is the number of features, and c is the number of partitions or clusters.

Clusters are described by their member objects and by their centers. Usually centroids are used as the centers of clusters. The centroid of each cluster is the point to which the sum of distances from all objects in that cluster is minimized. By using a partitioning clustering algorithm, $\boldsymbol{X}$ is partitioned into c clusters with a goal of obtaining low within-cluster and high between-cluster heterogeneity. That is, a cluster consists of objects which are as close to each other as possible, and as far from objects in other clusters as possible. Depending on research domains, dataset $\boldsymbol{X}$ is formed with data points that are the representations of objects which can be individuals, observations, cases, requirements, pixels etc.

While hard clustering algorithms like KM assign each object to exactly one cluster, soft partitioning or fuzzy clustering algorithms like FCM assign each object to different clusters with varying degrees of membership as mentioned above. In other words, while the membership to a cluster is exactly either 0 or 1 in KM it varies between 0 and 1 in FCM. Therefore, in the cases that we cannot easily decide that objects belongs to only one cluster, especially with the datasets having noises or outliers, FCM may be better than KM. For that reason, it is expected that KM algorithm may be a good option for exclusive clustering but FCM may give good results for overlapping clusters. In the following subsections KM and FCM is explained with their algorithmic steps.

2.1. K-means algorithm

KM iteratively computes cluster centroids for each distance measure in order to minimize the sum with respect to the specified measure. KM algorithm aims at minimizing an objective function known as squared error function given in Equation (1) as follows:

$$J_{KM}(\boldsymbol{X};\ \boldsymbol{V}) = \sum_{i=1}^{c}\sum_{j=1}^{n} D_{ij}^2 \tag{1}$$

Where,

D_{ij}^2 is the chosen distance measure which is generally in Euclidean norm: $\left\|x_{ij} - v_i\right\|^2$, $1 \leq i \leq c$, $1 \leq j \leq n_i$. Where n_i represents the number of data points in *i*th cluster.

For c clusters, KM is based on an iterative algorithm minimizing the sum of distances from each object to its cluster centroid. The objects are moved between clusters until the sum cannot be decreased any more. KM algorithm involves the following steps:

1) Centroids of c clusters are chosen from $\boldsymbol{X}$ randomly.
2) Distances between data points and cluster centroids are calculated.
3) Each data point is assigned to the cluster whose centroid is closest to it.
4) Cluster centroids are updated by using the formula in Equation (2):
$$\boldsymbol{v}_i = \sum_{i=1}^{n_i} x_{ij}\,/n_i\ ;\ 1 \leq i \leq c \tag{2}$$
5) Distances from the updated cluster centroids are recalculated.
6) If no data point is assigned to a new cluster the run of algorithm is stopped, otherwise the steps from 3 to 5 are repeated for probable movements of data points between the clusters.

2.2. Fuzzy C-means algorithm

FCM algorithm minimizes the objective function in Equation (3).

$$J_{FCM}(\boldsymbol{X};\ \boldsymbol{U},\boldsymbol{V})\ =\ \sum_{i=1}^{c}\sum_{j=1}^{n} u_{ij}^m\, D_{ij\boldsymbol{A}}^2 \tag{3}$$

This function differs from classical KM with the use of weighted squared errors instead of using squared errors only. In the objective function in Equation (3), $\boldsymbol{U}$ is a fuzzy partition matrix that is computed from dataset $\boldsymbol{X}$:

$$\mathbf{U} = \left[u_{ij}\right] \in M_{FCM} \tag{4}$$

The fuzzy clustering of $\boldsymbol{X}$ is represented with $\boldsymbol{U}$ membership matrix in $c \times n$ dimension.The element u_{ij} is the membership value of *i*th object to *j*th cluster. In this case, the *j*th column of $\boldsymbol{U}$ matrix is formed with membership values of n objects to *j*th cluster. $\boldsymbol{V}$ is a prototype vector of cluster prototypes (centroids):

$$\boldsymbol{V} = [\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_c], \mathbf{v}_i \in \mathbb{R}^p \tag{5}$$

$D_{ij\boldsymbol{A}}^2$ is the distances between *i*th features vector and the centroid of *j*th cluster. They are computed as a squared inner-product distance norm in Equation (6):

$$D_{ij\boldsymbol{A}}^2 = \left\|\mathbf{x}_j - \mathbf{v}_i\right\|_{\boldsymbol{A}}^2 = (\mathbf{x}_j - \mathbf{v}_i)^T \boldsymbol{A}(\mathbf{x}_j - \mathbf{v}_i) \tag{6}$$

In Equation (6), $\boldsymbol{A}$ is a positive and symmetric norm matrix. The inner product with $\boldsymbol{A}$ is a measure of distances between data points and cluster prototypes. When $\boldsymbol{A}$ is equal to $\boldsymbol{I}$, $D_{ij\boldsymbol{A}}^2$ is obtained in squared Euclidean norm. In Equation (3), m is a fuzzifier parameter (or weighting exponent) whose value is chosen as a real number greater than 1 ($m \in [1, \infty)$). While m approaches to 1 clustering tends to become crisp but when it goes to the infinity clustering becomes fuzzified. The value of fuzzifier is usually chosen as 2 in the most of applications. The objective function is minimized with the constraints as follows (7, 8 and 9):

$$u_{ij} \in [0,1];\ 1 \leq i \leq c, 1 \leq j \leq n \tag{7}$$

$$\sum_{i=1}^{c} u_{ij} = 1;\ 1 \leq j \leq n \tag{8}$$

$$0 <\ \sum_{j=1}^{n} u_{ij} < n;\ 1 \leq i \leq c \tag{9}$$

FCM is an iterative process and stops when the number of iterations is reached to maximum, or when the difference between two consecutive values of objective function is less than a predefined convergence value (ε). The steps involved in FCM are:

1) Initialize $\boldsymbol{U}^{(0)}$ membership matrix randomly.
2) Calculate prototype vectors: $\mathbf{v}_i = \frac{\sum_{j=1}^{n} u_{ij}^{m} \boldsymbol{x}_j}{\sum_{j=1}^{n} u_{ij}^{m}}$; $1 \leq i \leq c$ (10)
3) Calculate membership values with:
   $u_{ij} = \frac{1}{\sum_{k=1}^{c} \left(D_{ijA}/D_{kjA}\right)^{2/(m-1)}}$; $1 \leq i \leq c\,, 1 \leq j \leq n$ (11)
4) Compare $\boldsymbol{U}^{(t+1)}$ with $\boldsymbol{U}^{(t)}$, where t is the iteration number.
5) If $\left\|\boldsymbol{U}^{(t+1)} - \boldsymbol{U}^{(t)}\right\| < \varepsilon$ then stop else return to the step 2.

3. Datasets and parameters of the algorithms

We analyzed totally 10 datasets for comparing KM and FCM. While the datasets from DS1 to DS5 and DS8 were synthetically generated with a script developed by use of several R libraries. The remaining datasets were downloaded from the site of Speech and Image Processing Unit, School of Computing at University of Eastern Finland, FI (http://cs.joensuu.fi/sipu/datasets). (DS6 by Gionis *et al* (2007); DS7 by Zahn (1971); DS9 by Fu & Medico (2007); DS10 by Jain & Law (2005)). As shown in Figure 2 and 3, and listed in Table 1 the datasets from DS1 to DS4 consisted of equal sized rectangular, circular and ellipsoidal clusters spreading with a regular tiled pattern, the others were irregular shaped clusters spreading with regular and irregular patterns.

Table 1. Size and structure of the datasets

Dataset	*c*	*n*	n_c	Shape of clusters	Pattern
DS1	9	1800	200	Equal sized rectangles	Regular
DS2	9	1800	200	Equal sized circles	Regular
DS3	9	1800	200	Equal sized ellipses (normal, mid-eccentric)	Regular
DS4	9	1800	200	Equal sized ellipses (thin, high-eccentric)	Regular
DS5	14	1327	≈95	Different sized circles with some noises	Irregular
DS6	7	788	≈113	Different sized miscellaneous shapes	Irregular
DS7	6	399	≈66	Different sized miscellaneous shapes	Irregular
DS8	3	1200	400	2 concaves, 1 ellipse	Irregular
DS9	2	240	120	1 concave, 1 circle	Irregular
DS10	2	373	≈187	2 concaves	Irregular

In the synthetically generated datasets mentioned above, the inter-cluster variances have been set to a reasonable level to obtain well separated clusters. As listed in Table 1, in order to obtain dense structures for the synthetically generated datasets the cluster size (n_c) was set to 200 data points for each cluster in DS1, DS2, DS3, and DS4. The size of clusters varied between 66 and 400 with an average of 153 for the remaining datasets. All datasets were formed with two features ($p = 2$) for easy interpreting cluster structures via the scatter plots in two dimensional space.

The function `kmeans` from the `stats` package of R (R Core Team 2015) was used in KM clustering. It was run two times with the option of MacQueen method. The first run was for single pass of KM (KM-1) and the second was for 10 initial starts of KM (KM10). Since the cluster centers are randomly chosen before the start of iterations the clustering results can be different in each run of KM, so we randomly chose one of the result sets from several runs of KM1 for all datasets.

For FCM analysis we used the function `FKM` from `fclust` library developed by Ferraro and Giordani (2015) in R environment. `FKM` was run for only single random start. As one of the essential input arguments of `FKM` the fuzzifier was set to 2 (*m=2*) as a default value, and the convergence value

(ε) was set to ($1e - 09$). Squared Euclidean distance norm was used as the distance measure in both KM and FCM algorithms. As the c values, numbers of the clusters in the original datasets in Table 1 were used for both KM and FCM algorithms.

The performances of the algorithms were compared by using three criteria which were CPU time per iteration (TPI), CPU time per object (TPO), and the percentage of the objects moved out to other clusters from their original cluster after clustering. The CPU runtime required in each run of the algorithms was computed as the difference from `Sys.time()` with a precision of 9 digits which has been get before and after running the procedures. A notebook PC having i7 microprocessor and 8GB RAM was used for all type of analysis, and R was only active application during analysis.

4. Results and discussion

As shown in Figure 1 and listed in Table 2, KM with single start (KM1) required more iteration (ITRS) but less time per iteration (TPI) and less time per object (TPO) when compared to those obtained from KM with 10 starts (KM10). Except for the datasets DS6 and DS7, the numbers of iterations in KM1 were higher than in KM10 since the latter improved initial vectors of centroids with multiple starts. This finding indicates that running KM with multiple starts may give good clustering results with less number of iterations. On the other hand, TPIs of KM1 were averagely hundred to thousand times smaller than those of KM10. The similar trend was observed for TPOs. TPOs of KM1 were approximately thousand times smaller than those of KM10 as seen in Table 2. This advantage of KM1 does necessarily not mean that KM1 is superior to KM10 because of its relatively low clustering accuracy against KM10 which will be discussed later in this section.

Table 2. Computing time efficiency of the algorithms

DS #	KM1 iters	KM10 iters	FCM iters	KM1 tpi	KM10 tpi	FCM tpi	KM1 tpo	KM10 tpo	FCM tpo	% Inc. tpo
1	9	4	100	0.0003340	0.015260	0.187300	2.0e-06	3.4e-05	0.010406	30506
2	11	3	79	0.0001820	0.008005	0.226823	1.0e-06	1.3e-05	0.009955	76477
3	30	9	112	0.0000067	0.002668	0.199133	1.0e-06	1.3e-05	0.012390	95208
4	13	9	123	0.0000077	0.002780	0.212532	1.0e-06	1.4e-05	0.014523	103636
5	6	4	205	0.0001670	0.004253	0.241403	1.0e-06	1.3e-05	0.037293	286769
6	7	12	177	0.0002860	0.001918	0.080337	3.0e-06	2.9e-05	0.018045	62124
7	7	10	122	0.0001430	0.000050	0.048623	3.0e-06	1.3e-05	0.014867	114262
8	13	7	99	0.0001540	0.000286	0.061179	2.0e-06	2.0e-06	0.005047	252250
9	5	5	120	0.0000400	0.000060	0.012116	4.0e-06	1.3e-05	0.006058	46500
10	3	3	54	0.0003320	0.003002	0.020477	3.0e-06	2.4e-05	0.002964	12250

TPIs and TPOs from FCM algorithm for all datasets were extremely higher when compared to those from both KM1 and KM10. The percent of TPO increase from KM10 to FCM (in the last column of Table 2) revealed that a remarkably more execution time as much as several hundred thousand times were required by FCM algorithm.

As shown in Figure 1, KM10 and FCM required more iterations for the datasets consisting of non-regularly scattering clusters. While the highest number of iterations were obtained for DS6 with KM10, DS5 with FCM, and DS3 with KM1. KM1 showed approximately same iteration performance for all datasets excluding DS3.

As expected, higher execution times were needed for larger datasets (Column 2 of Figure 1) in spite of some exceptions. It was an interesting result that both CPU times and TPI values from KM1 and KM10 were the highest for the dataset DS1 although it consisted of same sized and well separated clusters like the clusters in DS2, DS3 and DS4. However we did not obtain the same result from FCM analysis in which the highest TPI and TPO values were for DS5. On the larger datasets FCM did not run faster than KM as claimed by Sheshasayee and Sharmila (2014), contrary it was remarkably slower in all datasets. However we observed that TPIs were higher for larger datasets in both KM10 and FCM. In FCM analysis we also determined that TPOs of larger datasets tends to be higher compared those of smaller datasets with some exceptions (i.e. DS6, DS7) for the datasets having non-regularly scattering clusters.

The results revealed that the algorithms showed special behaviors for TPIs against the scattering of clusters in datasets. TPIs from KM1 increased for the datasets whose clusters scattering with irregular patterns with an exception for DS1. In contrast to this finding, probably due to larger size of clusters we observed that TPIs from KM10 and FCM on the datasets having regularly scattering patterns were higher than those of the dataset whose clusters scattering with irregular patterns. The dataset DS5 had the highest TPO and TPI when analyzed with FCM. Since this dataset has the highest number of clusters (*c=14*) we understand that as the number of cluster increases the time complexity of FCM increases rapidly. So, we conclude that runtimes of FCM are mainly affected by the number of clusters rather than their sizes and shapes. But, for a generalized understanding of this finding, further studies should be carried out in the future.

Table 3 presents the number of member losing clusters (NCML), the number of objects moved to other clusters (NMO), and the percentage of objects moved to other clusters (PMO) by running the algorithms.

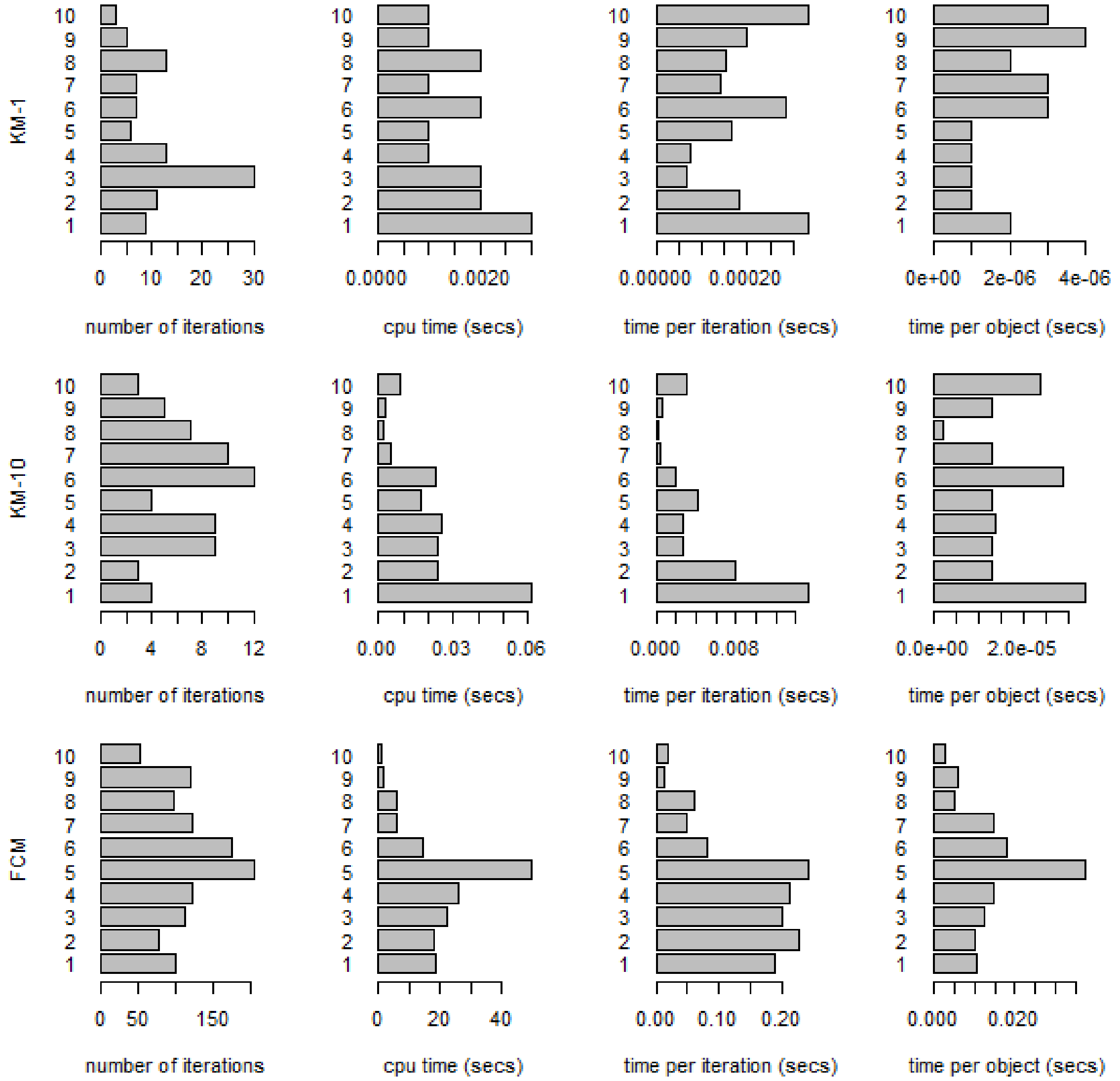

Figure 1. Number of iterations and runtimes per iteration and runtimes per object from the algorithms

Table 3. Clustering success of the algorithms

DS #	KM1 ncml	KM1 nmo	KM1 pmo	KM10 ncml	KM10 nmo	KM10 pmo	FCM ncml	FCM nmo	FCM pmo	Best Clustering Algorithms
1	3/9	186	10.33	0/9	0	0.00	0/9	0	0.00	KM10, FCM
2	2/9	191	10.61	0/9	0	0.00	0/9	0	0.00	KM10, FCM
3	1/9	1	0.00	1/9	1	0.00	3/9	17	0.01	KM1, KM10, FCM
4	5/9	214	11.88	0/9	0	0.00	4/9	18	0.01	KM10, FCM
5	6/14	79	5.95	7/14	23	1.73	8/14	22	1.65	KM10, FCM
6	3/7	148	18.78	3/7	125	15.86	4/7	154	19.54	-
7	5/6	146	36.59	5/6	137	34.34	4/6	95	23.81	-
8	2/3	13	0.01	2/3	9	≈0.01	2/3	15	≈0.01	KM1, KM10, FCM
9	2/2	41	17.08	2/2	39	16.25	2/2	36	15.00	-
10	2/2	80	21.45	2/2	80	21.45	2/2	84	22.52	-

In order to compare the accuracies of clustering, these values (NCML, NMO and PMO) can be used as the failure rate (or error rate) of the tested clustering algorithms. If an original/natural cluster loses its member objects this means that the used algorithm does not work properly. Therefore the NCML values in Table 3 can be used as the indicators of the failure rate (or success rate as '*1 minus failure rate*'). However NCMLs give roughly an idea about failure/success in comparison of the algorithms they will not be reliable measures since clusters may lose their members in varying degrees. For instance, while a cluster may lose only one member another may lose half of its members. Since the NCMLs will be equal for two cases, their usage may not be acceptable for comparing the failure or success of the algorithms. A better option in comparison of failure/success performances of the algorithms is to use the PMOs, percentages of objects moved from their original clusters. If a PMO value is 0 we can infer that the clustering algorithm finds the cluster perfectly. On the other hand, when it is increased clustering result cannot be seen well. In this paper we used a failure rate of 5% as an acceptable threshold value in the comparison of the performances of the algorithms.

As seen from PMOs in Table 3 and the scatter plots in Figure 1, KM10 and FCM had same success to find rectangular and circular clusters scattering with regular patterns in datasets from DS1 to DS4. KM1 was also surprisingly successful with its 0% of PMO for the dataset DS3 containing ellipsoidal clusters. As reported in many studies FCM gives the better results for circular clusters but not well for ellipsoidal clusters. However its PMO for DS3 was under 5% of acceptable level, it also gave higher PMO when compared to PMO of KM10 (or KM1) in this study. For the dataset DS4 which consisted of thin ellipsoidal clusters while KM1 was bad KM10 and FCM were good in favor of KM10 with zero failure rate. However KM10 and FCM were equally efficient to find clusters in the all datasets having the clusters scattering with regular patterns, KM10 was superior to FCM when the computing cost was also concerned as a privilege factor in the choice of an appropriate algorithm.

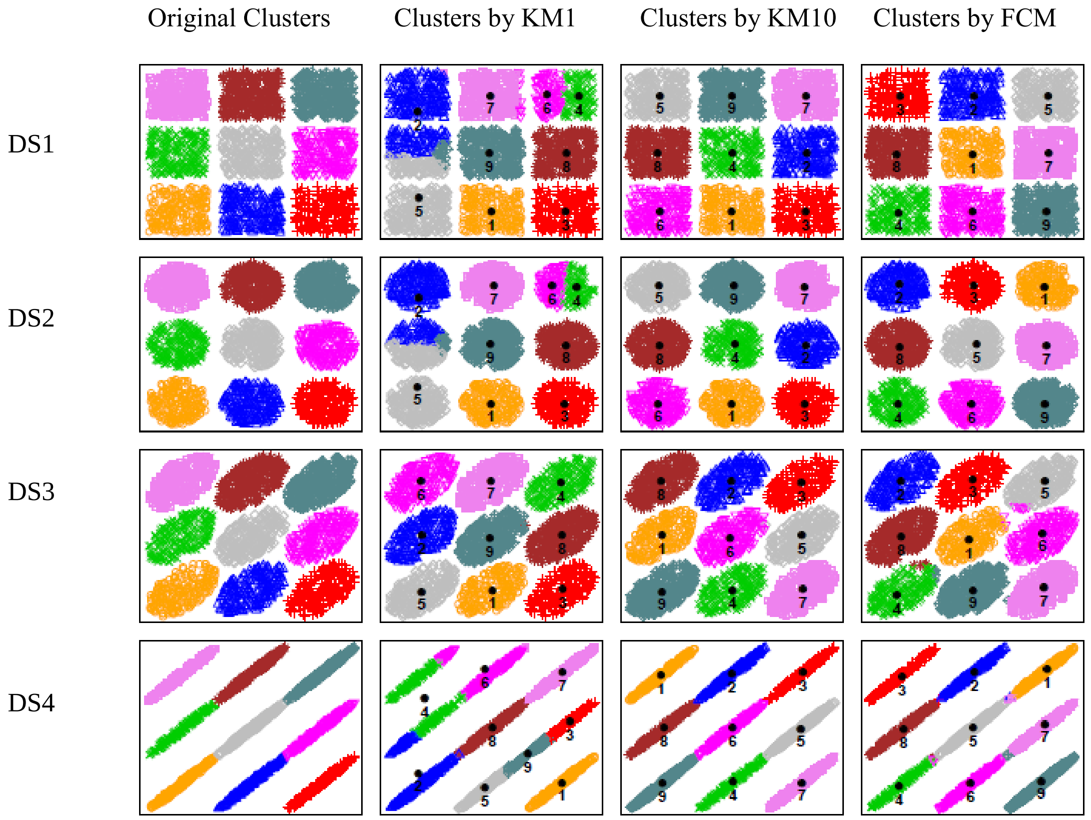

Figure 2. KM and FCM clustering for different shaped clusters scattering with regular patterns

As shown in Table 3 and Figure 3, neither KM1 and KM10 nor FCM could find the clusters in all datasets which are scattering with irregular or non-linear patterns except DS5 and DS8. For DS5 having not well separated clusters, KM10 and FCM showed similar success with the 1.73% and 1.65% of PMO respectively (Table 3). Therefore, KM with multiple starts or FCM can alternatively be used to handle good clustering results on the datasets containing circular shaped clusters even they non-linearly scattered. In spite of their nearly equal performances, we recommend that KM can be used for its lower computing time cost as will be seen in Table 2 and higher number of correctly found clusters. Similar conclusions were reported by Madhukumar & Santhiyakumari (2015). For DS8 which is formed with two concave clusters and one circular cluster in middle of them, the failure rates were approximately 0.01 by all algorithms compared in this paper.

Since DS6 and DS7 have the more complex structures the failure rates were higher than those of other datasets. Thus KM and FCM algorithms did not give the results which are above 5% of acceptable clustering failure level. Finally, for DS9 and DS10 having only two clusters, none of the algorithms gave acceptable clustering result. As shown in Table 3 the PMOs of the algorithms on irregularly scattering datasets (DS6, DS7, DS9 and DS10) were also higher than acceptable failure level. Consequently, we understood that KM and FCM algorithms are not good clustering options to partition datasets containing nested clusters scattering with irregular pattern as seen in Figure 3.

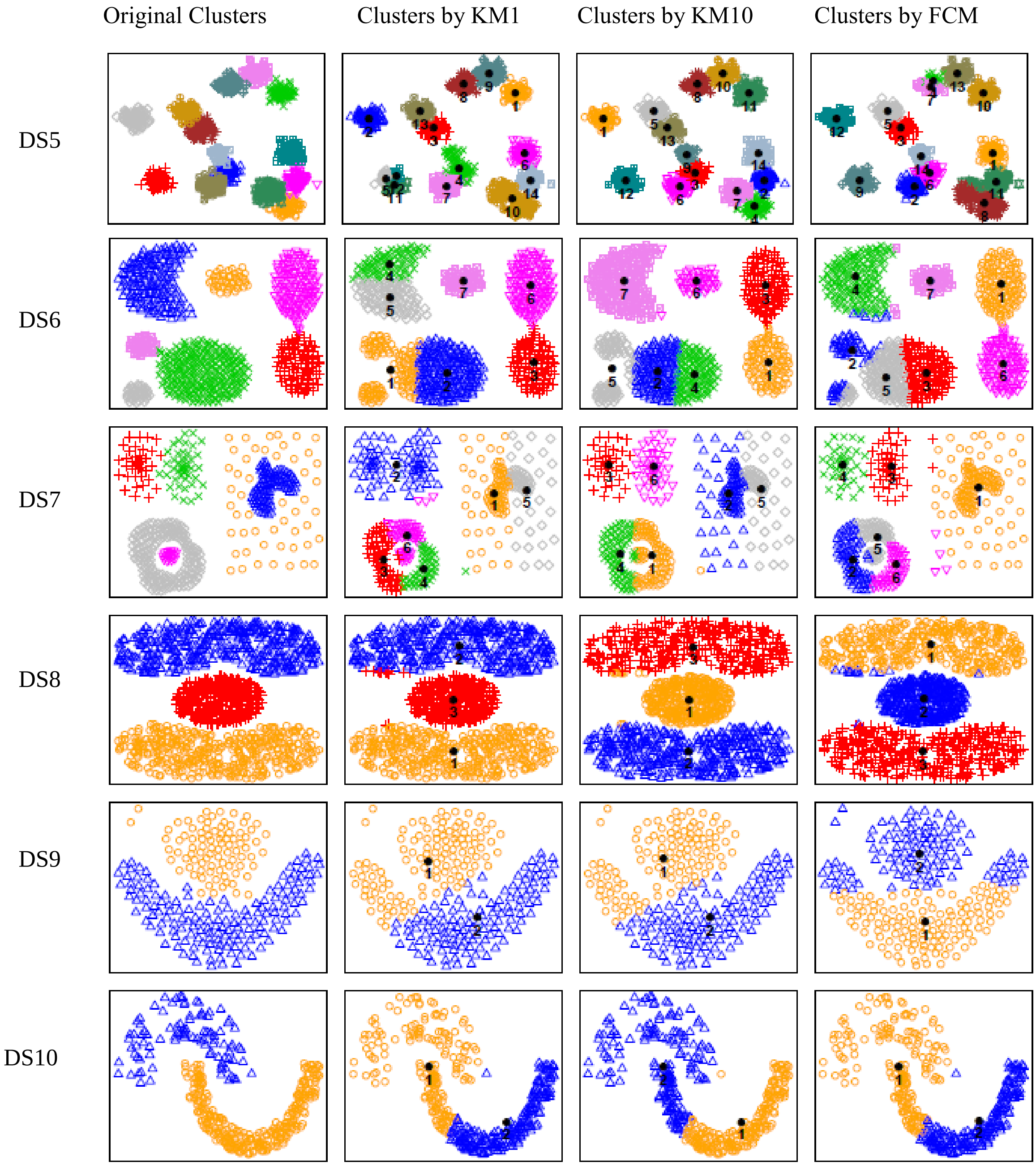

Figure 3. KM and FCM clustering for different shaped clusters scattering with non-linear patterns

5. Conclusions

KM was always extremely faster than FCM in all datasets containing the clusters scattering in regular or irregular patterns. FCM is an algorithm based on more iterative fuzzy calculations, so its execution was found comparatively higher as it is expected. Similar results were reported by Panda *et al.* (2012) for Iris, Wine and Lens datasets; by Jipkate & Gohokar (2012) for segmentation of images; by Ghosh & Dubey (2013) for Iris dataset; by Bora & Gupta (2014) for Iris dataset; by Sivarathri & Govardhan (2014) for diabetes data; and by Madhukumar & Santhiyakumari (2015) for brain MR images data.

An important factor in choosing an appropriate clustering algorithm is the shape of clusters in datasets to be analyzed. The clustering failure of FCM and KM10 was found nearly equal for all shapes of clusters scattering with a regular pattern. However their performances were better for circular and rectangular clusters when compared to ellipsoidal clusters, KM10 was relatively good. Further experimental studies should be conducted to clarify this finding by using other forms of distance norms like Manhattan and by applying the derivative algorithms of KM and FCM. According to a study by

Borgelt & Kruse (2005) regularized and constrained clustering is so robust that it can even be used without an initialization by the FCM algorithm with shape constraints. Testing the proposed approaches on real data with different ellipsoidal shapes of clusters may be helpful for a precise decision between the algorithms.

Sivarathri & Govardhan (2014) revealed that FCM is better than KM in term of accuracy of clusters on the diabetes dataset obtained from the UCI repository. However, in our study, neither KM nor FCM were successful to find the concave and other kind of arbitrary shaped clusters when they are not well separated. In the analysis of this kind of data structures we recommend that shape sensitive clustering algorithms should be used. For instance, the spectral clustering and hierarchical agglomerative methods for nested circular cluster structures; Ward, hierarchical agglomerative methods and density based methods such as Dbscan and Birch for concave clusters may be good options in cluster analysis. On the basis of experimental results, we recommend the use of KM with multiple starts because of its lower computational time than that of FCM algorithm for all shapes and well separated scattering clusters. As reported in many studies while FCM will give better results for noisy clustered datasets KM will be good choice for large datasets because of its execution speed. Thus, the use of KM should be a good starting point for large agricultural datasets due to its fast execution time.

As a final conclusion, there is no any algorithm which is the best for all cases. Thus, the datasets should be carefully examined for shapes and scatter of clusters in order to decide for a suitable algorithm. To achieve this, 2D and/or 3D scatter plots of datasets provide good idea to understand the structure of clusters in datasets. When multi-featured objects are analyzed, in order to overcome to plot for multidimensional space, a dimension reduction technique such as multidimensional scaling (MDS) or principal components analysis (PCA) can be applied to reduce dimensions of datasets. Moreover, by using a suitable sampling method this process can be completed in shorter execution times.

References

Ali, MA, Karmakar, GC & Dooley, LS 2008 ‘Review on Fuzzy Clustering Algorithms’. IETECH Journal of Advanced Computations, vol. 2, no. 3, pp. 169 – 181.

Bezdek, JC 1981, Pattern Recognition with Fuzzy Objective Function Algorithms, Plenum Press, New York., doi: 10.1007/978-1-4757-0450-1

Bora, DJ & Gupta, AK 2014 ‘A Comparative study Between Fuzzy Clustering Algorithm and Hard Clustering Algorithm’. Int. J. of Computer Trends and Technology, vol. 10, no. 2, pp. 108-113.

Borgelt, C & Kruse, R 2005 ‘Fuzzy and Probabilistic Clustering with Shape and Size Constraints’. Proc. of the 11th Int. Fuzzy Systems Association World Congress (IFSA'05, Beijing, China), pp. 945-950.

Di Martino, F, Loia, V & Sessa, S 2007 ‘Extended Fuzzy C-means Clustering Algorithm for Hotspot Events in Spatial Analysis’. Int. J of Hybrid Intelligent Systems, no. 4, pp. 1–14.

Di Martino, F & Sessa, S 2009 ‘Implementation of the Extended Fuzzy C-Means Algorithm in Geographic Information Systems’. J. of Uncertain Systems, vol. 3, no. 4, pp. 298-306.

Dong, W, Ren, JD & Zhang, D 2011. Hierarchical K-Means Clustering Algorithm Based on Silhoutte and Entropy. H.Deng et al. (Eds): AICI 2011, Part I, LNAI vol. 7002, pp. 339-347. Springer-Verlag Berlin, Heidelberg., doi: 10.1007/978-3-642-23881-9_45

Dunn, JC 1973 ‘A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters’. J. of Cybernetics, vol.3, no.3, pp. 32-57., doi: 10.1080/01969727308546046

Ferraro, MB & Giordani, F 2015 ‘A Toolbox for Fuzzy Clustering Using the R Programming Language’. Fuzzy Sets and Systems (in press), doi: 10.1016/j.fss.2015.05.001

Fu, L & Medico, E 2007, ‘FLAME, a Novel Fuzzy Clustering Method for the Analysis of DNA Microarray Data’. *BMC Bioinformatics*, vol. 8, no. 1, pp. 3.

Ghosh, S & Dubey, SK 2013 ‘Comparative Analysis of K-Means and Fuzzy C-Means Algorithms’. Int. J. Advanced Computer Science and Applications, vol. 4, no.4, pp. 35-39.

Gionis, A, Mannila, H & Tsaparas, P 2007 ‘Clustering Aggregation’. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 1, no.1, pp. 1-30.

Jain, A & Law, M 2005 'Data Clustering: A User's Dilemma'. *Lecture Notes in Computer Science*, 3776, pp. 1-10., doi: 10.1007/11590316_1

Jipkate, BR & Gohokar, VV 2012 'A Comparative Analysis of Fuzzy C-Means Clustering and K Means Clustering Algorithms'. Int. J. of Computational Engineering, vol. 2, no. 3, pp. 737-739.

Kaur, M & Kaur, U 2013 'Comparison Between K-Mean and Hierarchical Algorithm Using Query Redirection'. Int. J. of Advanced Research in Computer Science and Software Engineering, vol. 3 , no. 7, pp. 1454-1459.

MacQueen, JB 1967 'Some Methods for Classification and Analysis of Multivariate Observations'. Proc. of 5th Berkeley Symp. on Mathematical Statistics and Probability, Berkeley, University of California Press, pp. 281-297.

Madhukumar, S & Santhiyakumari, N 2015 'Evaluation of K-Means and Fuzzy C-means Segmentation on MR Images of Brain'. The Egyptian J. of Radiology and Nuclear Medicine, vol. 46, no. 2, pp. 475–479., doi: 10.1016/j.ejrnm.2015.02.008

Panda, S, Sahu, S, Jena, P & Chattopadhyay, S 2012 'Comparing Fuzzy-C Means and K-Means Clustering Techniques: A Comprehensive Study'. Advances in Intelligent and Soft Computing, vol. 166, pp. 451-460, doi: 10.1007/978-3-642-30157-5_45

R Core Team 2015 'R: A Language and Environment for Statistical Computing'. R Foundation for Statistical Computing, Vienna, Austria. URL http://www.r-project.org.

Ramesh, D & Vardhan, BV 2013 'Data Mining Techniques and Applications to Agricultural Yield Data'. Int. J. of Advanced Research in Computer and Communication Engineering, vol. 2, no. 9, pp. 3477-3480.

Sheshasayee, A & Sharmila, P 2014 'Comparative Study of Fuzzy C-means and K-means Algorithm for Requirements Clustering'. Indian J. of Science and Technology, vol. 7, no 6, pp. 853–857.

Sivarathri, S & Govardhan, A 2014 'Experiments on Hypothesis Fuzzy K-Means is Better Than K-Means for Clustering' Int. J. Data Mining & Knowledge Management Process, vol. 4, no. 5. pp. 21-34., doi: 10.5121/ijdkp.2014.4502

Suganya, R & Shanthi, R 2012 'Fuzzy C- Means Algorithm - A Review'. Int. J. of Scientific and Research Publications, vol. 2, no. 11, pp. 1-3.

Tiwari, M & Misra, B 2011 'Application of Cluster Analysis in Agriculture - A Review Article'. Int. J. of Computer Applications, vol. 36, no.4, pp. 43-47.

Velmurugan, T 2012 'Performance Comparison between K-Means and Fuzzy C-Means Algorithms Using Arbitrary Data Points'. Wulfenia Journal, vol. 19, no. 8, pp. 234-241.

Zahn, CT 1971 'Graph-theoretical Methods for Detecting and Describing Gestalt Clusters'. *IEEE Transactions on Computers*, vol. 100, no. 1, pp. 68-86., doi: 10.1109/t-c.1971.223083