Over-Segmentation of Feature Space for Initialisation of K-means Clustering

Ursani Ahsan Ahmad, Kidiyo Kpalma, Joseph Ronsin

Abstract – **This paper suggests over-segmenting the feature space for improved initialisation of the k-means clustering method. Two over-segmentations (OS) of a dataset are achieved and then fused to determine the best-possible initial means to perform the iterative process of k-means. The method is tested over both, image as well as non-image data. The image data is segmented using texture features based on Gabor energy features (GEF). The method is compared with single-run k-means and k-means with multiple restarts. Encouraging results have been obtained while segmenting a variety of texture images and the non-image data. The 40% OS is found to give the best results. This technique addresses the issue of data clustering in all the possible areas including data mining, pattern recognition, machine learning and statistics.**

Index Terms -- Fusion, K-means Clustering, Over-segmentation, Data mining

I. INTRODUCTION

Despite continued research by hundreds of researchers engaged for many long years, data clustering remains one of the most non-trivial and one of the most challenging tasks in the fields of data mining, statistics, bio-informatics, and image segmentation alike. Although there are many clustering techniques, however, so far k-means remains the most popular among unsupervised techniques of data clustering due to its simplicity and faster convergence. Nevertheless, often falling into a local minimum remains a problem semper instans. The k-means is generally performed several times (usually hundreds of times) before choosing the best solution among the several ones found therein. Moreover one is never sure that the best solution found is the optimal one.

This paper presents a method of initialising k-means clustering technique, herein called as FOOS, which helps prevent k-means from falling in the suboptimal points that lead to extremely undesirable clustering; thereby achieving optimal or near-optimal solution. A good comparison of the initialisation methods for k-means is [1] that discusses the problems associated with the k-means clustering.

The remainder of the paper is organized as follows: section II describes the 3 methods used and compared herein, section III describes more precisely the problems and conditions of failure of k-means and the success of FOOS, section IV estimates the computational cost involved in the clustering methods, section V presents the results and their comparison and finally section VI concludes with final remarks on the results and possible future work.

II. METHODS

Here is the explanation of the three methods of clustering used and compared herein, namely, single-run k-means and FOOS algorithms implemented with our own programs and k-means with multiple restarts implemented with the software *Gene Cluster 3.0*.

A. *K-Means Algorithm*

K-means clustering, invariant to instance order, based on the similarity metric of Euclidean distance is performed with the random initialisation. The dataset is randomly sampled to a much smaller subset, and then K means are selected randomly, making sure that there is a (considerable) minimum allowable distance between any two of the randomly selected means.

The iterative process of clustering is then performed to determine new class means and assign a class to each data-point in the dataset on the basis of closest Euclidean distance until difference between the current and the previous class means is zero. Generally, it is known that k-means assumes that the desired clusters are populated in spherical Gaussian distributions; and that any deviation from this situation causes k-means algorithm to fall into a sub-optimal point.

B. *Gene Cluster 3.0*

Gene Cluster 3.0 is a computer program developed by M. de Hoon [2] based on the original Cluster program written by Mike Eisen of Berkley Lab. In addition to the results obtained with our program of single-run k-means, results obtained with *Gene Cluster 3.0* are also presented and compared with those from FOOS. *Gene Cluster 3.0* performs k-means several times before choosing the best solution out of the several available. The criterion for choosing the best solution is the minimum Intra-Cluster Distance [1] defined as

Manuscript received on July 22, 2007.

All the authors are affiliated with INSA de Rennes, France, where A. A. Ursani (Corresponding author) is currently a PhD student. Phone: 33223238747. His mother institute is Mehran University of Engineering & Technology (MUET), Jamshoro, Sindh, Pakistan, where he is serving as Assistant Professor. MUET is sponsoring his stay at INSA de Rennes. Email: aursani@ens.insa-rennes.fr, kidiyo.kpalma@insa-rennes.fr, joseph.ronsin@insa-rennes.fr,

$$E = \sum_{k=1}^{K} \sum_{n=1}^{N_k} \left| \mu_k - x_n \right| , \qquad (1)$$

where K is the total number of clusters, N_k is cardinality of the kth cluster, μ_k is the mean feature vector of the kth cluster, and x_n is the feature vector of the nth data member of a running cluster k.

C. The Proposed Approach

The proposed approach, referred to as Fusion Of Over-Segmentations "FOOS", is a means of finding the initial means that, unlike k-means, leads to the optimal or a near-optimal solution even if the desired clusters do not lie in the spherical Gaussian distribution. FOOS performs the k-means procedure only 3 times: it initialises the 3rd run of k-means with the cluster means found from the 2 preceding runs of the algorithm.

The clustering algorithm suggested here finds out 2 over-segmentations (OS) and then performs a kind of fusion of the clusters found therein. The k-means algorithm is first used to segment a dataset into M and N clusters, respectively, where K is the desired number of clusters, M>K and N=M+1. The 2 OS are then fused to determine good initial means to perform simple k-means clustering explained in the preceding section. Hence those K pairs of clusters, one from each M-cluster and N-cluster OS, respectively, are selected that have the largest number of common member data-points.

Least possible OS is suggested for the datasets with K≤5. In this case, M=K+1. The procedure is described mathematically in (2) through (7).

$$\bigcup_{m=1}^{M} X_m = \bigcup_{n=1}^{N} Y_n = I , \qquad (2)$$

where M=K+1, N=K+2, I is the set of all the data-points in the dataset, X_m is one of the M clusters found in first OS and Y_n is one of the N clusters found in the 2nd OS.

$$\forall m, n, \exists Z_{m,n} = X_m \cap Y_n , \qquad (3)$$

where $Z_{m,n} \in Z$, $\|Z\| = M \times N$ and $\|*\|$ represents the cardinality of a set.

$$T_{m,n} = \left\| Z_{m,n} \right\| \text{ m=1, 2, .., M and n=1, 2, …, N} \qquad (4)$$

$$\begin{aligned} &\forall m, \exists W_m = (m,n) : \max(\overset{N}{\underset{n=1}{T_{m,n}}}) \wedge \\ &(m,n) : \min(\max(\overset{N}{\underset{n=1}{T_{m,n}}})) \notin W \Big\| \|W\| = k \end{aligned} \qquad (5)$$

$$\vec{\mu} = \frac{1}{T_{W_n}} \sum_{i=1}^{T_{W_n}} \vec{x}_{W_n,i} \quad \forall \mathrm{I} \mid 1 \le \mathrm{I} \le \mathrm{k} \wedge \mathrm{I} \in N, \qquad (6)$$

where W_n is the nth order pair in the set W that is the set of order pairs representing the K largest clusters of common data-points from M and N segmentations; $\vec{\mu}_n$ and T_{W_n} , respectively, are mean and cardinality of the nth cluster, and $\vec{x}_{W_n,i}$ is the feature vector of the ith data-point in the nth cluster

Equation (5) explains that the smallest sets are rejected, selecting only the K largest clusters among $Z_{m,n}$. FOOS algorithm then computes the corresponding K means from them to be considered as initial means to perform the following and decisive iterative process of k-means over the dataset as follows:

$$x_j \in X_n \leftrightarrow d_{j,n} = \min \overset{k}{\underset{i=1}{(d_{j,i})}} \qquad (7)$$

where X_n is the nth cluster, $d_{j,n}$ is the Euclidian distance of x_j from the nth cluster, and $d_{j,i}$ is the Euclidian distance of x_j from ith cluster.

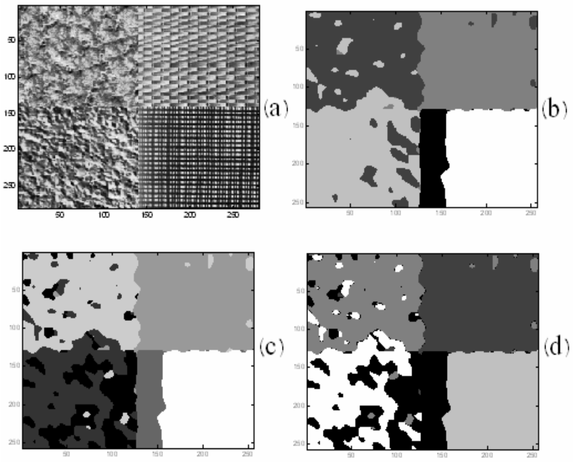

Fig.1. (a) The image with four textures from Brodatz collection, Segmentation into (b) 5 clusters (c) 6 clusters (d) The 4 largest classes of common member pixels

For example, fig. 1 shows an image with 4 (K) textures D92, D55, D4, and D21 from the Brodatz album, its segmentation into 5 (M), 6 (N), and 4 (K) clusters having the highest number of common member pixels. The pixels that are not part of the 4 largest classes of common members are shown in black. Table 1 shows the statistics of the number of common pixels in this particular case of the image in fig. 1. The table shows that class 3 from M segmentation and class 4 from N segmentation have the highest number of common member pixels, i.e. 16324. The following 3 (M,N) order pairs are (2,5) with 13964, (5,6) with 12777, and (4,2) with 11135 common members. These sets of common members form the new clusters with the new means that are better means to be taken as initial means to perform k-means for clustering the dataset into K segments.

For the datasets having more classes, i.e. K>>5, the least possible OS doesn't seem to be the best solution, as shown in the results in section V. In case where K≤5, we perform least possible OS leading to M=K+1=6, and N=K+2=7; i.e. 2 more on 5 or 40% OS. If we extend this idea of 40% OS to a dataset with K=10, the 2 proposed OS shall be M=13 and N=14.

Table 1: Statistics of the clusters formed after OS of the image in 5 clusters (left column) and 6 clusters (top row). The other cells in the table show the number of common pixels found in the 2 OS, with the highest 4 numbers in bold

	Segmentation into N (k+2) 6 clusters							
	Class #	n	1	2	3	4	5	6
	m	$\|Y_n\|$	7395	11245	3656	16388	14075	12777
		$\|X_n\|$ $T_{m,n}$						
Segmentation Into M (k+1) 5 clusters	1	3660	13	0	3647	0	0	0
	2	16710	2598	96	0	52	**13964**	0
	3	16405	38	10	0	**16324**	33	0
	4	15975	4746	**11135**	5	12	77	0
	5	12786	0	4	4	0	1	**12777**

III. Notion Behind FOOS

Reference [3] is a recent work that points out 3 established problems/disadvantages with the k-means algorithm that are already pointed out by many earlier works. First that it requires the number of clusters beforehand, second that it is sensitive to the outliers, and third that any two randomly chosen initial means might be too close to be considered as two distinct centroids. Reference [1] notes that despite having all the advantages of convergence and computational simplicity, the k-means is affected by the choice of initial means and may easily converge into a local optimum due to its assumption that the clusters it is trying to find lie in a spherical Gaussian distribution.

Requiring the number of clusters beforehand is not always a problem. Often, the user prefers to decide upon it. Even the results of manual clustering of any given dataset will yield different outcomes since the number of clusters is almost always subjective depending on how rigorous segmentation is required. Therefore in many situations, the number of clusters (K) is a user-defined parameter. It is hence rather an advantage in many situations. The random selection of initial means can be repeated if any two of the selected means do not lie at a considerable distance. Rest of the problems, i.e. its assumption of spherical Gaussian distribution, sensitivity to outliers, and dependence on the initial means, all are inter-related. Keeping in view the determination of the number of clusters using Gaussian separation in [4], one can summarize these problems into a single problem statement as follows:

A given dataset usually has a different number of clusters from the Gaussian analysis [4] viewpoint than the number of clusters determined by application, requirement, and/or subjective assessment. This disagreement makes distributions of the desired clusters look non-Gaussian and some of their members as outliers to the k-means algorithm, causing sensitivity to the choice of initial means while dividing the dataset into the desired number of non-Gaussian clusters.

The problem of so-called *outliers* is the same problem of non-Gaussian distribution in the different words. For a given dataset, segmenting in to a larger number of clusters relieves the problem of *outliers* or that of skewness in the cluster distributions; since more and smaller/finer Gaussian classes can more closely approximate non-Gaussian distributions. References [5-7] explain how non-Gaussian distributions can be approximated by piecewise Gaussian distributions.

Consider a situation requiring a dataset to be segmented into 2 clusters, whereas the dataset seems to have 2 major clusters (large in population) “A”, “B”, and a minor (small in population) cluster “C” as per Gaussian separation [4]. The 2 major populations lie close in the feature space in such way that the minor cluster “C” is more distinct from cluster “A” than the major cluster “B” is from “A”. In this situation, the k-means is most probably going to result in a merged class comprising the 2 major clusters “A” and “B” and the minor cluster “C” as a separate class. Even the k-means with multiple restarts will result in the same solution, since the merger of the 2 major classes will result in the minimum Intra-Cluster Distance.

Doing OS helps distinguish 2 relatively closer major classes from one another and the minor class. The following fusion process excludes the smaller/minor class, finding out the best initial means from the 2 major populations.

IV. Cost Analysis

The computational cost of k-means is given by

$$C_{kmeans} = O(KtN), \tag{8}$$

where K represents the number of clusters to be found, t is the number of iterations performed, and N is the cardinality of the dataset to be clustered.

Reference [8] argues that since $\{K, t\} << N$, the computational cost in (8) can be approximated by

$$C_{kmeans} = O(N). \tag{9}$$

Since FOOS can be thought of successive application of k-mans to perform clustering into K+2, K+1, and K clusters respectively, its computational cost can be computed as

$$C_{FOOS} = O((K + (K+1) + (K+2))tN) \tag{10}$$

and considering that *3Kt<<N*, can be approximated as

$$C_{FOOS} \cong O(3KtN) \approx O(N)\,, \tag{11}$$

which tells that the computational complexity of the two methods happens to be the same.

The conventional k-means with multiple restarts is obviously computationally more demanding than FOOS that takes only 3 runs, whereas in the case of k-means, it is a function of cardinality of the dataset.

V. RESULTS

The results on both, the image and the non-image data are presented from the k-means, the *Gene Cluster 3.0*, and the FOOS algorithms.

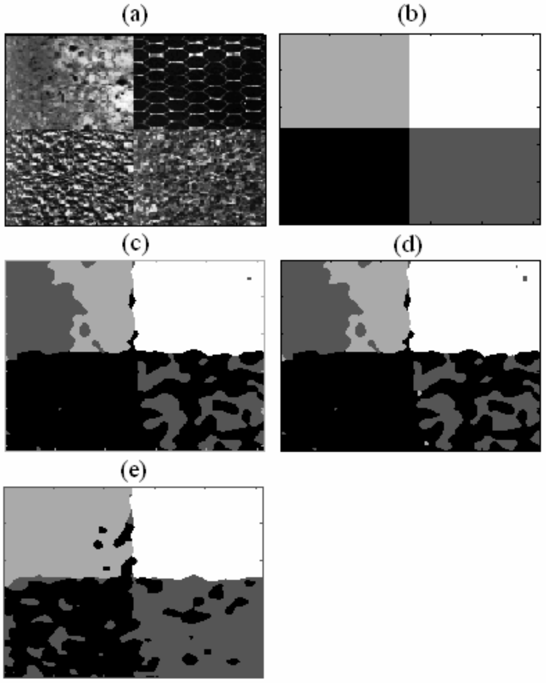

Fig.2. (a) The image with 4 textures from Brodatz collection, (b) its ideal segmentation, segmentation result with (c) k-means, (d) Gene Cluster 3.0, (e) FOOS algorithm

A. Clustering the Image Data

For image data, the results are presented for the application of texture segmentation. The texture segmentation is one of the important aspects of not only the computer vision systems but also finds application in remote sensing (RS). The texture segmentation is carried out by extracting texture features and then using some data clustering technique such as k-means to segment the areas having homogeneous textures.

i. Collages of Four Brodatz Textures

An image with collage of 4 patches of the Bordatz textures D73, D34, D57, and D29, each of 128^2 pixels from Brodatz texture collection is shown in fig. 2(a). In this case, we over-segment the image into M=5 and then into N=6 clusters.

The 2 OS are then fused to find 4 clusters. As shown in fig. 2(c), the k-means heavily misclassifies the textures on the bottom-right (D29) and the top-left (D73) corners achieving the accuracy of 69.5%. *Gene Cluster 3.0* is no different from k-means, achieving even worse, i.e. 69.2% accuracy. Contrarily, the FOOS algorithm considerably outperforms with an accuracy of 88.8% and is quite successful in discriminating all the classes.

In fact, many such combinations of textures were segmented. Table 2 lists the details of the collages and shows the results obtained with the k-means, FOOS, and the *Gene Cluster 3.0*. The *Gene Cluster 3.0* doesn't provide the optimal solution after making 100s of runs of k-means, and sometimes gives even worse results than k-means performed once by our program.

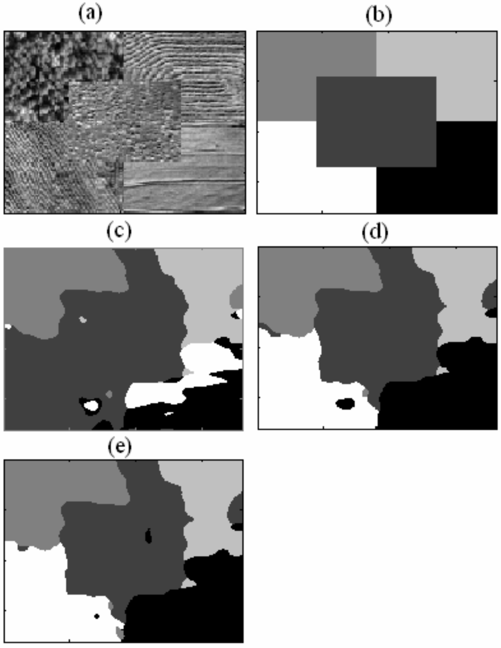

Fig.3. (a) The image with 5 textures from airborne CASI sensors, (b) its ideal segmentation, segmentation result with (c) k-means, (d) Gene Cluster 3.0, (e) FOOS algorithm

ii. Five Textures From a Natural RS Scene

A collage of 5 textures from an RS image taken from an airborne sensor called CASI is presented. Fig. 3(c) shows that the k-means fails in discriminating between bottom-left texture and the central texture, causing a forceful creation of the 5th class from within the region of homogeneous texture on the bottom-right corner. Contrary to this, the FOOS segments all the textures successfully with most of the misclassifications in the top-right texture that has been confused with the central texture as shown in fig. 3(e). The result of *Gene Cluster 3.0*, shown in fig. 3(d) is slightly worse than FOOS, achieving 89.1% accuracy after 26 runs of k-means. K-means achieves 65.8% accuracy as compared to 89.4% accuracy with FOOS.

In this example, k-means (single run) took 43 iterations, whereas FOOS took a total of 93 iterations that is far below 3 times those of k-means.

Table 2: The list and details of the images with a collage of 4 Brodatz textures each: The last column shows how many times the Gene Cluster found clusters using k-means before choosing the best among them

S. No.	Textures Selected (From top left to bottom right)	Segmentation Accuracy			
		K-Means	FOOS (3 runs of k-means)	Gene Cluster 3.0	
				Accuracy	No. of Runs
1	D73, D34, D57, D29	69.5%	88.8%	69.2%	274
2	D4, D55, D9, D21	67.6%	97.0%	96.9%	178
3	D24, D84, D4, D21	67.8%	95.4%	95.4%	453
4	D92, D55, D4, D21	67.5%	81.1%	67.2%	029
5	D3, D22, D112, D80	60.4%	91.5%	91.4%	192
6	D54, D84, D57, D100	52.3%	69.8%	69.5%	028
7	D73, D37, D57, D29	55.0%	73.2%	55.0%	050
Overall Average Accuracy		62.8%	85.3%	77.8%	172

iii. A Natural Image from Berkley Database

Here are the results on a natural image from Berkley Segmentation Database (BSD) reported in [9]. As shown in fig. 4(a), the image apparently has 3 textures, 1st one that of the grass, 2nd one that of the zebras, and 3rd in the rest of the image. The image has the size of 481X321. Fig. 4(b) illustrates its segmentation (into 36 segments) by a human subject (user #1107) as given in the BSD benchmark, clearly outlining the zebras, the grass, and rest of the image (the non-grass background). It should be noted however that the provided segmentation is with reference to the edge detection only, and not with respect to the texture. Fig. 4(c) and 4(d) show the result obtained with k-means and *Gene Cluster 3.0*, respectively, with the outline of fig. 4(b) overlaid on them. Fig. 4(e) shows the result obtained with FOOS. The *Gene Cluster 3.0* performed 472 k-means processes and took hours to complete.

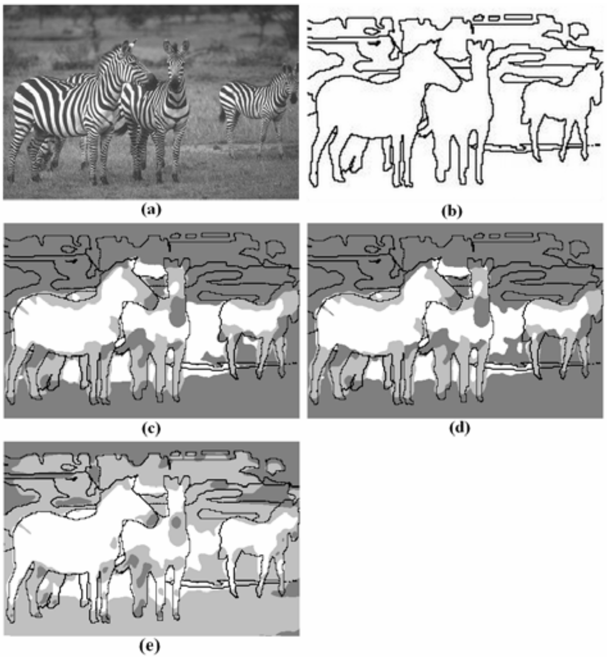

Fig.4. (a) The zebra image from BSD, its segmentation (b) as provided by BSD, (c) with k-means, (d) with *Gene Cluster 3.0*, (e) with FOOS

FOOS successfully separates the zebras, the grass, and the non-grass background. Having no texture, the mouths and the feet of the zebras also appear different from the zebra skin. Remaining misclassifications apparently indicate the reduced discrimination ability of the extracted features. On the other hand, the k-means and *Gene Cluster 3.0* fail in distinguishing between the grass and the non-grass background, forcefully creating a class within the zebra texture and classifying legs as different from zebra's main body. The k-means and *Gene Cluster 3.0* also misclassify the non-occluded part of the otherwise occluded zebra's back.

B. Clustering the Non-Image Data

Results are presented on 2 datasets from UCI machine learning repository [10].

As seen in the results on the image data, the single run k-means may result in very good or very bad clustering solution depending on its random initialisation. Hence, results from single-run k-means are considered no more.

i. A Dataset with 2 Classes

This is a database of human body dimensions, whose original source is [11]. The data include 507 observations of 21 body dimension as well as age, weight, height, and sex. The subjects include 247 male and 260 female individuals. The feature of sex was removed from this dataset. Further, all the feature values were rescaled from -250 to 250 to ensure that all the feature vectors get the equal weight in the distance measurement. The dataset was segmented into 2 clusters corresponding to the 2 sexes. The removed column of sex was used as the ground truth. The results from *Gene Cluster 3.0* and FOOS are exactly the same, i.e. 87.4%, with 443 instances classified correctly. Very small number of classes makes the task quite easy; therefore results ought to be the same. However, *Gene Cluster 3.0* performed 9216 runs of the k-means before reaching the solution, against 3 runs by FOOS.

ii. A Dataset with 10 Classes

This database was originally provided by E. Alpaydin and C. Kayna. The dataset contains 2000 instances corresponding to hand-written digits “0” to “9”. The class distribution is uniform, viz. 200 instances of each class. The number of dimensions is 649. *Gene Cluster 3.0* stopped after performing just a single run of k-means, and as suggested, it was therefore run repeatedly 4 times. Each time, it stopped after performing a single run of k-means, yielding accuracies of 80.45%, 80.15%, 80.55%, and 80.60%, respectively. FOOS was performed 4 times with different values of M, as illustrated in table 3.

Table 3: FOOS Results with different over-segmentations

M	N	% Over-segmentation	Accuracy
11	12	20%	75.95%
12	13	30%	83.15%
13	14	40%	91.70%
14	15	50%	91.15%

The last 3 results, i.e. corresponding to 30-50% over-segmentations are better than those of *Gene Cluster 3.0*. The best result is the one corresponding to 40% over-segmentation, with an accuracy of 91.7%, i.e. 11% more than that of *Gene Cluster 3.0*.

Reference [12] reports that using neural network classifiers (NNC), even the learning error was not less than 10%, which corresponds to accuracies of under 90%! It further states that the combined NNC failed in distinguishing between class “6” and class “9”. The clusters resulting from FOOS show that class “6” is well separated from class “9” with not a single member of class “6” classified as belonging to class “9” or vice versa; although both cluster “6” and cluster “9”, have some misclassifications, with more than 90% accuracies.

VI. Conclusion

To initialize the iterative process of k-means clustering, an approach based on the fusion of over-segmentations is proposed. The approach is shown to be successful with the segmentation of various image as well as non-image data that is publicly available. The favourable comparison is also shown with the single-run k-means and *Gene Cluster 3.0*, reinforcing the theoretical notions behind FOOS presented in section III.

Since FOOS makes only 3 runs of simple k-means, its computational complexity is also much lower than the other proposed methods that require performing k-means exhaustively or repetitively on the multivariate data. Table 2 shows that *Gene Cluster 3.0* gives the same or worse results after ~172 runs as compared to FOOS that is equivalent to 3 runs of k-means.

To test its repeatability, the FOOS was performed many times repeatedly. It gave exactly the same results in some cases and only slightly different results in others. For example, it was always exactly the same result for the natural image of fig. 4.

Computationally, FOOS is found to take less iterations than 3 times those taken by a single run of k-means, ensuring that it is almost never more expensive than 3 runs of k-means.

Least possible OS is tried with the datasets containing up to 5 clusters. On the other hand, for a dataset with 10 classes, rather 40% OS is found to be the best.

Acknowledgment

The authors are thankful to Professor Kacem Chehdi and his research team TSI2M at ENSSAT Lannion for granting permission to use the remote sensing images from airborne CASI sensors shown in fig. 3. We are also thankful to all the contributors in making available the datasets used herein.

Lastly, A. A. Ursani thanks Mehran University of Engineering & Technology, Jamshoro, Sindh, Pakistan for financial support for his research work at INSA.

References

[1] J. A. Lozano, J. M. Pena, P. Larranaga. An empirical, comparison of four initialization methods for the k-means, algorithm. Pattern Recognition Letters 20:1027–1040, 1999.

[2] M. J. L. de Hoon, S. Imoto, J. Nolan, and S. Miyano. Open Source Clustering Software. *Bioinformatics*, 20 (9):1453-1454, 2004.

[3] Bernard Chen, Phang C. Tai, R. Harrison, and Yi Pan. Novel Hybrid Hierarchical-K-means Clustering Method for Microarray Analysis, Proceedings of IEEE Computation Systems Bioinformatics Conference Workshops, 2005

[4] Chang J.H., Fan K.C. and Chang Y. L. Multi-modal gray-level histogram modeling and decomposition. Image and Vision Computing (20), 203-216, 2002.

[5] Alexander Steinwolf. Software for kurtosis and skewness fitting by way of piecewise-Gaussian approximation, Bulletin of the International Statistical Institute, Contributed Papers of 49th ISI Session, Florence, Book 2, p. 433-434, 1993.

[6] Alexander Stienwolf. Approximation and Simulation of Probability distributions with a variable kurtosis value. Computational Statistics and Data Analysis (21):163-180, Elsevier,1996

[7] Alexander Stienwolf and Stephen A. Rizzi. Non-Gaussian Analysis of Turbulent Boundary Layer Fluctuating Pressure on Aircraft Skin Panels. Jrnl. of Aircraft, 43(6):1662-1675, 2006

[8] Eduardo R. Hruschka, Estevam R. Hruschka, Thiago F. Covoes, Nelson F. F. Ebecken. Feature Selection for Clustering Problems: a Hybrid Algorithm that iterates between k-means and a Bayesian filter. Proc. the International Conference on Hybrid Intelligent Systems (5):405 – 410, 2005

[9] Martin and C. Fowlkes and D. Tal and J. Malik. A Database of Human Segmented Natural Images and its Application to Evaluating Segmentation Algorithms and Measuring Ecological Statistics, Proc. 8th Int'l Conf. Computer Vision (2):416-423, 2001 http://www.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/segbench/BSDS300/html/dataset/images/gray/253027.html

[10] Murphy, P. M. and Aha, D. W. UCI Repository of machine learning databases [http://mlearn.ics.uci.edu/MLSummary.html] Irvine, CA: Univ. of Calif., Dept. of Information & Computer Sc., 1994.

[11] Grete Heinz, Louis J. Peterson, Roger W. Johnson, and Carter J. Kerk. Exploring relationships in body dimensions. Journal of Statistics Education, 11(2), 2003

[12] M. van Breukelen and R.P.W. Duin, Neural Network Initialization by Combined Classifiers. Proc. 14th Int. Conf. Pattern Recognition (1):215, 1998