A Parameter Free Clustering Algorithm

Omar Kettani
Scientific Institute,
Physics of the Earth Laboratory
Mohamed V- University
Rabat, Morocco

Faical Ramdani
Scientific Institute,
Physics of the Earth Laboratory
Mohamed V- University
Rabat, Morocco

ABSTRACT
In data mining, most of clustering algorithms either require that the user provides in advance the exact number of clusters, or to tune some input parameter, which is often a difficult task. The present paper intends to overcome this problem by proposing a parameter free algorithm for automatic clustering. We evaluated its performance by applying on several benchmark datasets. Experimental results demonstrated that the proposed approach is effective.

General Terms
Data Mining, Clustering , Algorithms.

Keywords
Parameter free, automatic clustering, agglomerative clustering.

1. INTRODUCTION
In data analysis, clustering consists of grouping a given dataset into a predefined number of disjoint sets, called clusters, so that the elements in the same cluster are more similar to each other and more different from the elements in the other cluster. Most of existing clustering algorithms depend on one ore more tuning parameters, which are often difficult to tune, because they may require many empirical error-trials computations before obtaining satisfactory results. The most prominent clustering algorithm is k-means [1]. Given a set of n data points (objects) $X = \{x_1 , \ldots , x_n\}$ in R^d and an integer k, the clustering problem consists to determine a set of k centroids

$C = \{c_1 , \ldots , c_k\}$ in R^d , so as to minimize the following Sum of Square Error (SSE) function:

$$SSE=\sum_{x\in D} \min_{i=1,...,k} \| x - c_i \|^2$$

where $\| . \|^2$ denotes the Euclidean norm. The basic k-means is a greedy algorithm which has two stages: Initialization, in which we set the seed set of centroids, and an iterative stage, called Lloyd's algorithm [1]. Additionally, Lloyd's algorithm has two steps: The assignment step, in which each object is assigned to its closest centroid, and the centroid's update step. The time required for the assignment step is O(nk), while the centroid's update step and the computation of the error function is O(n).

K-means algorithm has a major drawback: the user must specify in advance the correct number of clusters, which is usually a difficult task when the distribution of the given data set is unknown.

In this paper, an alternative parameter free method for automatic clustering is introduced. It is based on the Agglomerative Clustering Method (ACM) proposed by the authors in a previous work [2]. Algorithm validation is conducted using several real-world and artificial clustering data sets from the UCI Machine Learning Repository [3].

In the next section, some related work are briefly discussed. Then the proposed approach is described in Section 3. Section 4 presents experimental results of this approach on different standard data sets and reports its performance. Finally, in Section 5 we draw conclusions and suggest some directions for future research.

2. RELATED WORK
Despite the fact that finding an optimal number of clusters k for a given data set is an NP-hard problem [4], several method have been developed to find k automatically.

Pelleg and Moore [5] proposed the X-means algorithm, which proceed by learning k with k-means using the Bayesian Information Criterion (BIC) to score each model, and chooses the model with the highest BIC score. However, this method tends to overfit when it deals with data that arise from non-spherical clusters. Tibshirani et al. [6] introduced the Gap statistic, which compares the likelihood of a learned model with the distribution of the likelihood of models trained on data drawn from a null distribution. This method is suitable for finding a small number of clusters, but has difficulty when k increases.Hamerly and Elkan [7] proposed the G-means algorithm, based on K-means algorithm, which uses projection and a statistical test for the hypothesis that the data in a cluster come from a Gaussian distribution. This algorithm works correctly if clusters are well-separated, and fails when clusters overlap and look non-Gaussian. Density based clustering is to discovers clusters of arbitrary shape in spatial databases. The DBSCAN algorithm [8] requires two parameters: ε (Eps) and the minimum number of points required to form a cluster (minPts). Usually, it is difficult to find these optimal parameters, because many empirical attempts are required before to get good quality results.

In the present work, an alternative approach is proposed, aiming to overcome this issue.

3. PROPOSED APPROACH
The proposed algorithm starts by setting $k=\text{floor}((n)^{1/2})$, where n is the number of objects in the given data set. This choice is motivated by the fact that this number lies in the range from 2 to $(n)^{1/2}$, as reported by Pal and Bezdek in [9]. Then in a first phase, it applies the ACM method proposed by the authors in. In the second phase, the two clusters having the closest centroids are merged. At each iteration, the maximum of CH cluster validity index (Calinski and Harabasz [10]) of the current partition is stored. We used this index because it is relatively inexpensive to compute, and it generally outperforms other cluster validity indices as reported by Milligan and Cooper in [11]. This process is repeated until k=2. Finally, the algorithm outputs the optimal k and partition corresponding to the maximum value of CH stored so far. This algorithm is outlined in the pseudo-code below:

Algorithm PFACM

Input: X= $\{x_1, x_2, \ldots, x_n\}$ in R^d

Output: k mutually disjoint clusters $C_1, \ldots, C_k$

such that $X=\bigcup_{j=1}^{k} C_j$

$k \leftarrow \lceil (n)^{1/2} \rceil$

[I,c]□ ACM(X,k)

ko □k

Io □ I

CHo □ CH(I)

While k>2 **do**

j□ argMin(□C_i □)

i<=k

c_j□ []

k□k-1

if CHo <CH(I) **then**

ko □k

Io □ I

CHo □ CH(I)

end if

end while

Output: ko and Io

The pseudo-code of ACM is outlined in the pseudo-code below:

Algorithm ACM(X,k)

Input: X and k

Output: k mutually disjoint clusters $C_1, \ldots, C_k$

such that $X=\bigcup_{j=1}^{k} C_j$

1 for i=1:k

m_i□X_i

C_i□ □X_i□

X□X-□X_i□

end for

2 compute D□$(d(m_i,m_j))_{1 \le i \ne j \le k}$ mu□Min(D) and (a,b)□Arg(Min(D))

i,j i,j

i□k+1

3 while X≠∅

d_i□Min($d(X_i,m_j)$) and c□Arg(Min $d(X_i,m_j)$)

j j

if d_i<mu **then**

C_c□ C_c□ □X_i□

m_c□(□C_c□m_c+X_i)/(□C_c□+1)

D(c,:)□ $(d(m_c,m_j))_{1 \le j \le k}$

D(:,c)□D(c,:)'

else

C_a□ C_a∪C_b

m_a□(□C_a□m_a+□C_b□m_b)/(□C_a□+□C_b□)

D(a,:)□ $(d(m_a,m_j))_{1 \le j \le k}$

D(:,a)□D(a,:)'

C_b□ □X_i□

m_b □X_i

D(b,:)□ $(d(m_b,m_j))_{1 \le j \le k}$

D(:,b)□D(b,:)'

end if

X□X-□X_i□

i□i+1

mu□Min(D) and (a,b)□Arg(Min(D))

h,j h,j

end while

3.1 Complexity

The time complexity of the first phase is $O(n^{3/2})$, since the running time of ACM is O(nk) and $k=n^{1/2}$.

The second phase requires $n^{1/2}$ x$O(n^{1/2})$, since each iteration i requires $O(i^{1/2})$ operations to update the centroids distance matrix and $O(i^{1/2})$ operations to evaluate the CH index of the current partition. Thus, the overall time complexity of PFACM is $O(n^{3/2})$.

4. EXPERIMENTAL RESULTS

Algorithm validation is conducted using different data sets from the UCI Machine Learning Repository. We evaluated its performance by applying on several benchmark datasets and compare with k-means, once PFACM has found k, the number of clusters.

Silhouette index (Kaufman and Rousseeuw [12]) which measures the cohesion based on the distance between all the points in the same cluster and the separation based on the nearest neighbor distance, was used in these experiments in order to evaluate clustering accuracy. (bigger average silhouette value indicates a higher clustering accuracy). Experimental results are reported in table 1 and figure 1, and some clustering results are depicted in figure 2 to 6.

Table 1. Experimental results of PFACM application on different datasets in term of mean Silhouette value

Dataset	k	k found	K-means sil.	PFACM sil.
breast	2	2	**0.7542**	0.7294
iris	3	**3**	0.7542	**0.7786**
glass	7	15	**0.6914**	0.6009
ruspini	4	**4**	**0.9086**	**0.9086**
thyroid	2	7	**0.7520**	0.7194
wine	3	8	**0.5043**	0.4126
yeast	10	3	0.2995	**0.7701**
a1	20	**20**	0.7185	**0.7693**
a2	35	**35**	0.6998	**0.7734**
a3	50	**50**	0.6695	**0.7835**
D31	31	**31**	0.6871	**0.9220**
dim32	16	**16**	0.7042	**0.9962**
dim64	16	**16**	0.8506	**0.9985**
dim128	16	**16**	0.7430	**0.9991**
dim256	16	**16**	0.8216	**0.9996**
dim512	16	**16**	0.6947	**0.9997**
R15	15	**15**	0.7879	**0.9361**
Unbalance	8	**8**	0.8132	**0.9727**
s1	15	**15**	0.7173	**0.8783**
s2	15	**15**	0.6796	**0.7828**
s3	15	18	**0.6422**	0.5939
s4	15	71	0.5492	**0.5546**

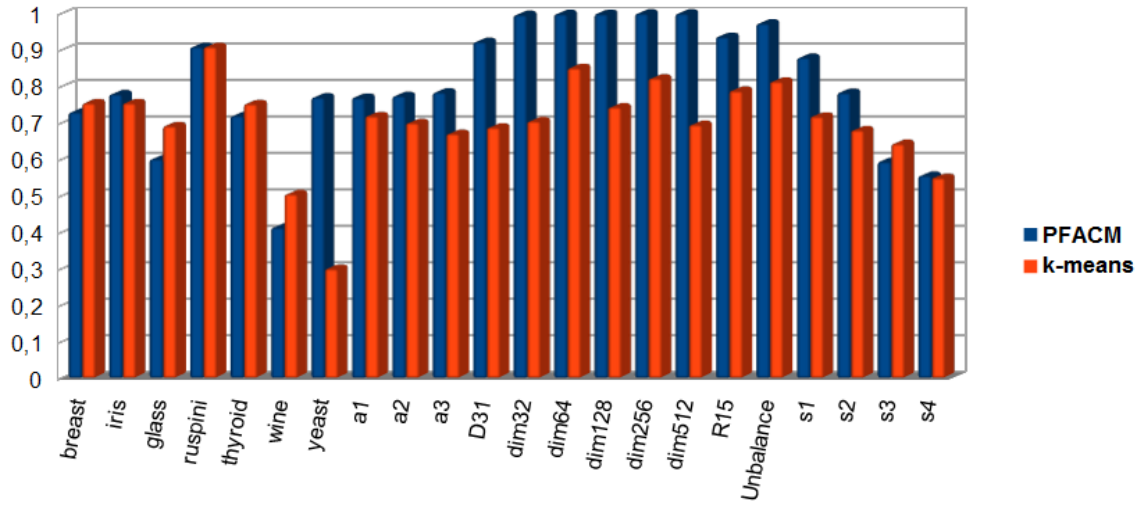

Fig 1: Chart of mean Silhouette index for both PFACM and k-means applied on different datasets.

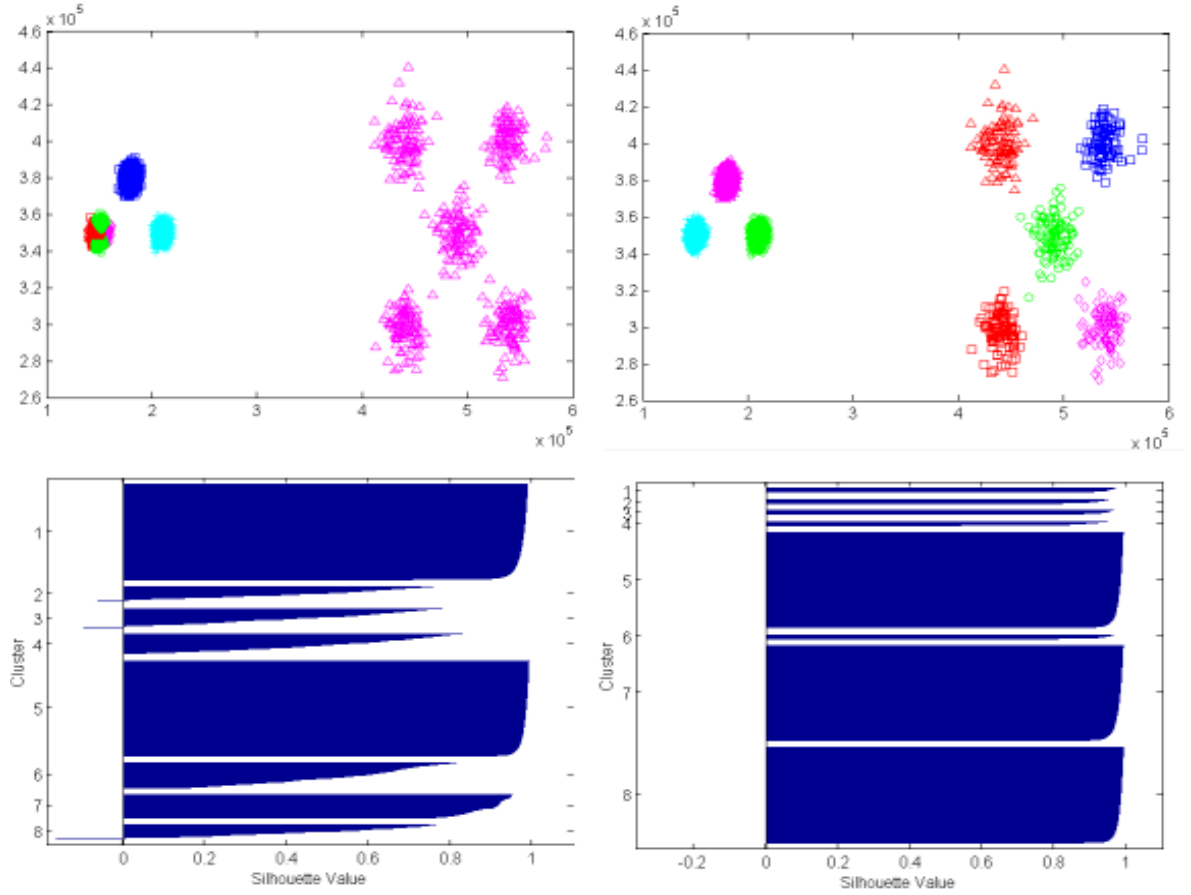

Fig 2: Clustering results of Unbalance dataset using k-means (on left) and PFACM (on right)

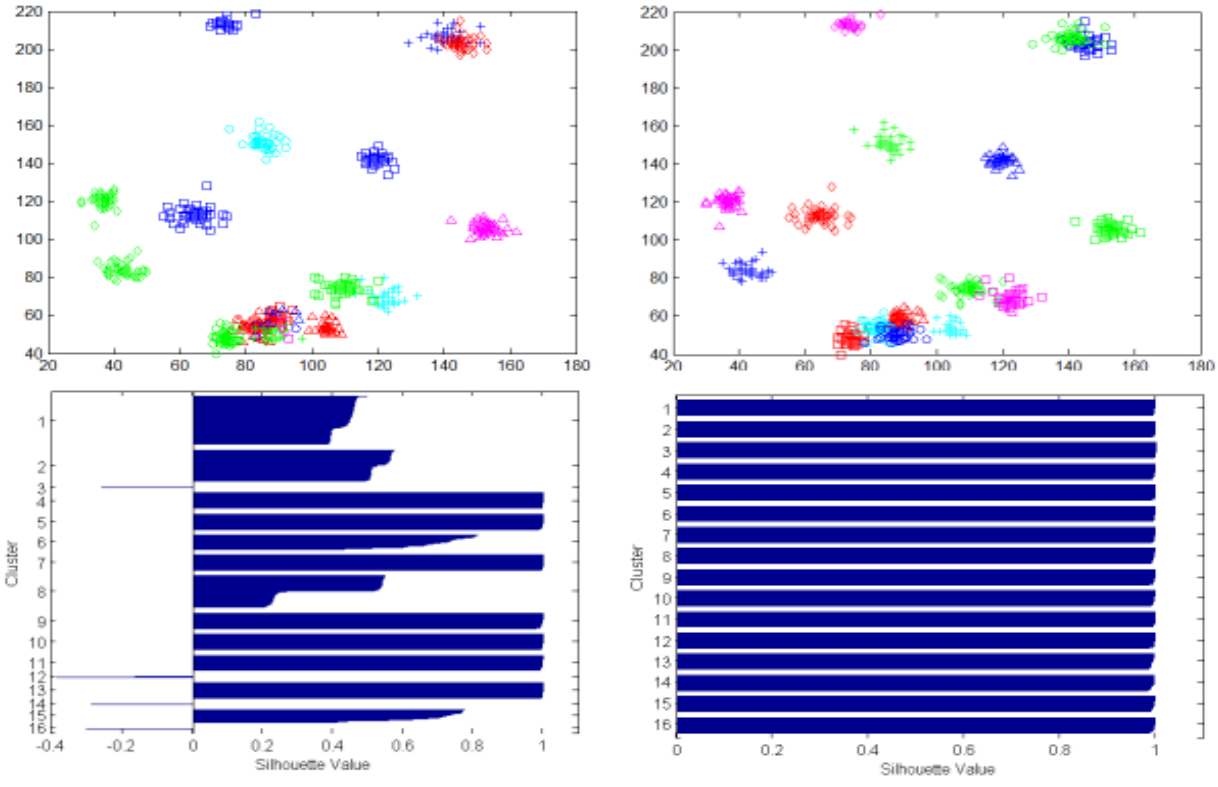

Fig 3: Clustering results of dim32 dataset using k-means (on left) and PFACM (on right)

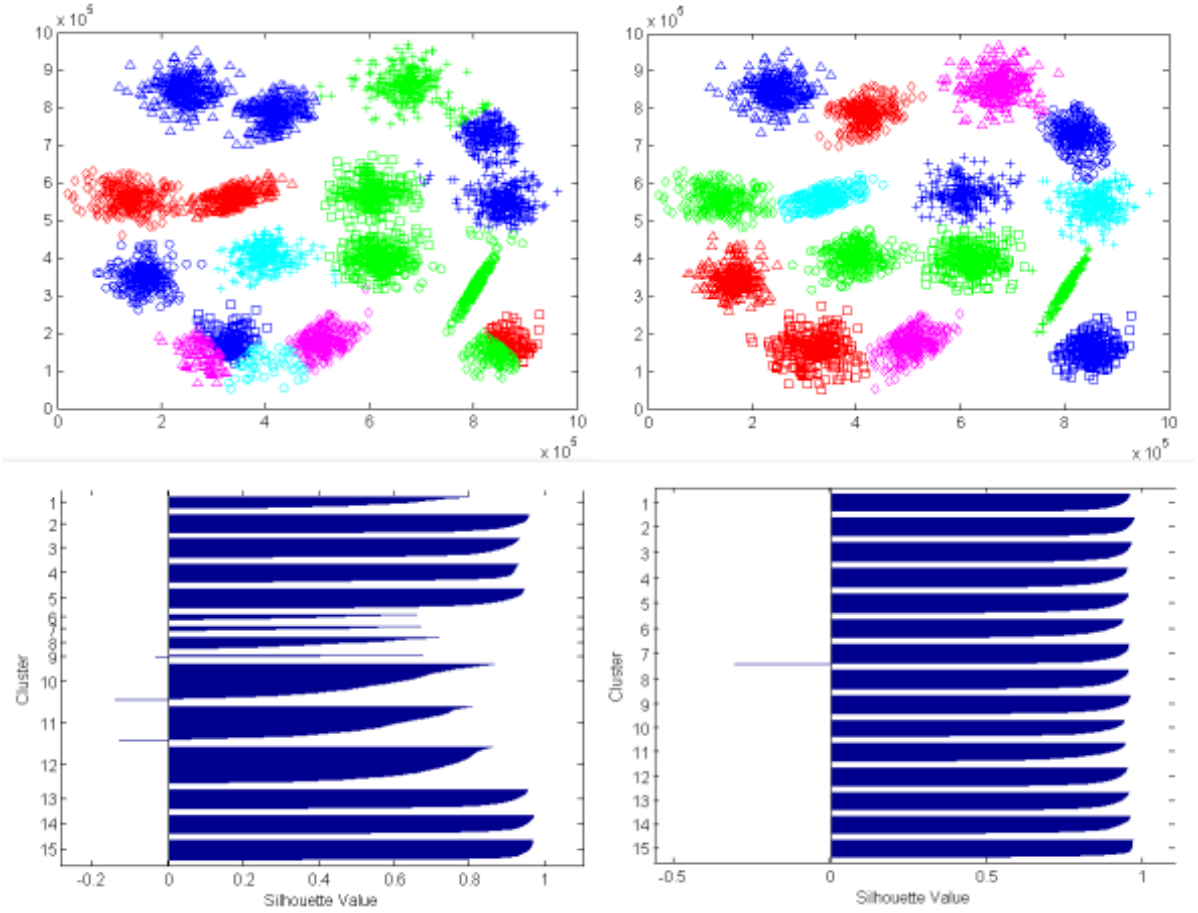

Fig 4: Clustering results of S1 dataset using k-means (on left) and PFACM (on right)

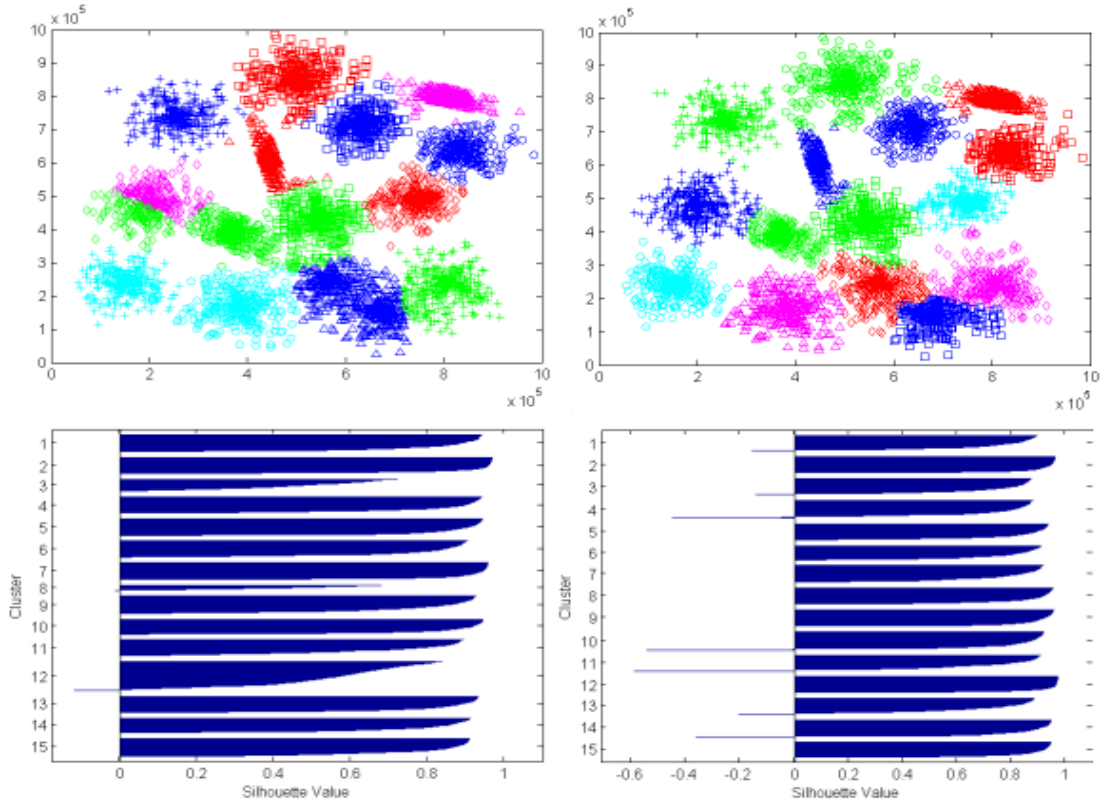

Fig 5: Clustering results of S2 dataset using k-means (on left) and PFACM (on right)

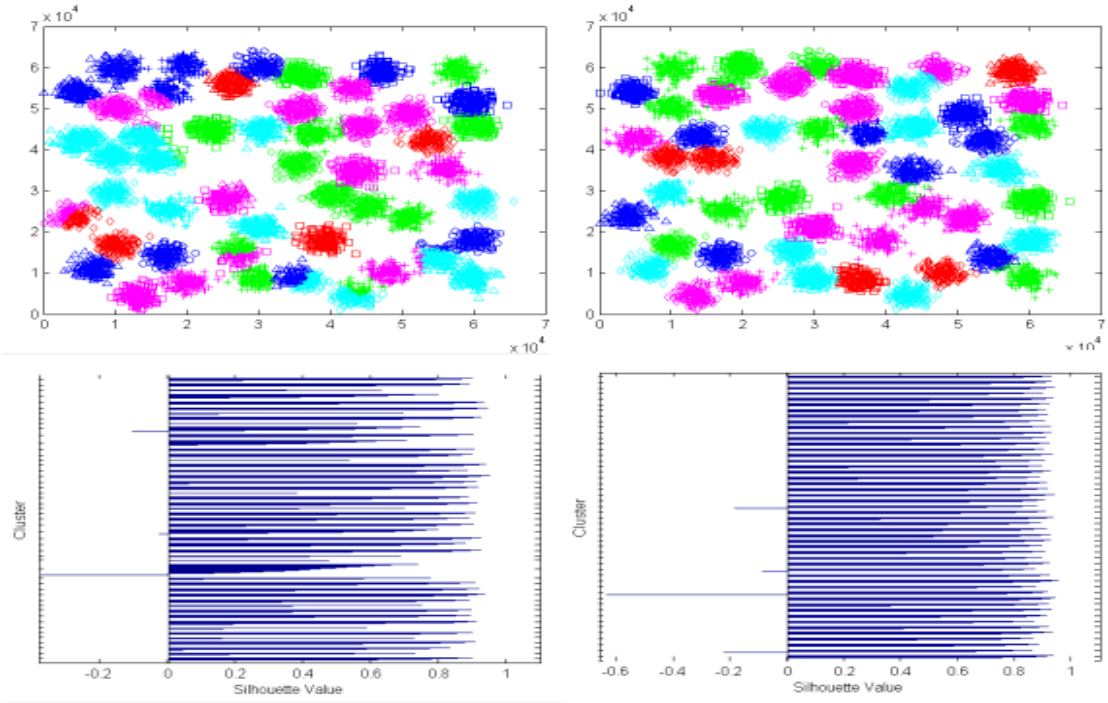

Fig 6: Clustering results of a3 dataset using k-means (on left) and PFACM (on right)

5. CONCLUSION

In this paper, an algorithm was suggested for automatic clustering. It is based on a simple deterministic clustering approach proposed by the authors in a previous work [2**]**. Experimental results demonstrated that this algorithm is able to find the appropriate number of clusters in almost all tested data sets. With this approach, non experts can expect good quality clusters without assistance from experts towards parameter tuning.

In future work, it will be of interest to find a tighter upper bound on the number of clusters, instead of $n^{1/2}$, in order to reduce the number of computations steps of the proposed approach. An other possible improvement will consist to try more adequate similarity measure instead of Euclidean distance, in order to enhance its clustering accuracy. Further research will explore these directions.

6. ACKNOWLEDGMENTS

Our thanks to the anonymous reviewers for their helpful comments.

7. REFERENCES

[1] Lloyd., S. P. (1982). "Least squares quantization in PCM". IEEE Transactions on Information Theory 28 (2): 129–137. doi:10.1109/TIT.1982.1056489.

[2**]** Kettani, O. ; Ramdani, F. & Tadili, B. An Agglomerative Clustering Method for Large Data Sets.International Journal of Computer Applications 92(14):1-7, April 2014. DOI:10.5120/16074-4952

[3] Asuncion, A. and Newman, D.J. (2007). UCI Machine LearningRepository [http://www.ics.uci.edu/~mlearn/MLRepository.html]. Irvine, CA: University of California, School of Information and Computer Science.

[4] Aloise, D.; Deshpande, A.; Hansen, P.; Popat, P. (2009). "NP-hardness of Euclidean sum-of-squares clustering". Machine Learning 75: 245–249. doi:10.1007/s10994-009-5103-0.

[5] Dan Pelleg and Andrew Moore. X-means: Extending k-means with efficient estimation of the number of clusters. In Proceedings of the 17th International Conf. on Machine Learning, pages 727–734. Morgan Kaufmann, 2000.

[6] Robert Tibshirani, Guenther Walther, and Trevor Hastie. Estimating the number of clusters in a dataset via the Gap statistic. Journal of the Royal Statistical Society B, 63:411–423, 2001.

[7] Greg Hamerly and Charles Elkan. Learning the k in k-means. In Proceedings of the seventeenth annual conference on neural information processing systems (NIPS), pages 281–288, 2003

[8] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in Proc. Second Int'l Conf. Knowledge Discovery and Data Mining (SIGKDD), 1996

[9**]** Pal, N.R. and Bezdek, J.C. (1995) On Cluster Validity for the Fuzzy c-Means Model. IEEE Transactions on Fuzzy Systems, 3, 370-379. http://dx.doi.org/10.1109/91.413225

[10] T. Calinski and J. Harabasz. A dendrite method for cluster analysis. Communications in Statistics, 3:1–27, 1974.

[11] G. W. Milligan and M. C. Cooper. An examination of procedures for determining the number of clusters in a data set. Psychometrica, 50:159–179, 1985.

[12] L. Kaufman and P. J. Rousseeuw. Finding groups in Data: "an Introduction to Cluster Analysis". Wiley, 1990.