An Agglomerative Clustering Method for Large Data Sets

Omar Kettani, Faycal Ramdani, Benaissa Tadili
LPG Lab.
Scientific Institute
Mohamed V University, Rabat-Morocco

ABSTRACT

In Data Mining, agglomerative clustering algorithms are widely used because their flexibility and conceptual simplicity. However, their main drawback is their slowness. In this paper, a simple agglomerative clustering algorithm with a low computational complexity, is proposed. This method is especially convenient for performing clustering on large data sets, and could also be used as a linear time initialization method for other clustering algorithms, like the commonly used k-means algorithm. Experiments conducted on some standard data sets confirm that the proposed approach is effective.

General Terms

Clustering, Algorithms.

Keywords

Agglomerative clustering, k-means initialization.

1. INTRODUCTION

Clustering is the process of grouping data into disjoint set called clusters such that similarities among data members within the same cluster are maximal while similarities among data members from different clusters are minimal. The optimization of this criterion is an NP hard problem in general Euclidean space d, even when the clustering process deals with only two clusters [1]. To tackle this problem, numerous approximation algorithms have been proposed, seeking to find near optimal clustering solution in reasonable computational time.

Clustering algorithms could be categorized into two major categories: partitional clustering, which determines all clusters at once, by dividing large clusters into small ones, and agglomerative clustering, which constructs a hierarchy of clusters by merging small clusters.

Agglomerative clustering process is generally slower than divisive clustering but allows more flexibility because it permits the user to supply any arbitrary similarity function defining what constitutes a similar cluster pair to merge together.

In this paper, an alternative agglomerative clustering method (called ACM), characterized by a low computational complexity, is introduced. This approach is particularly suitable for clustering massive data sets and has an easy implementation, without requiring any tuning parameter, except k, the number of clusters. Furthermore, it could also be used as a linear time initialization method for other clustering algorithms, like the k-means algorithm, in order to overcome one of its main drawbacks: its sensitivity to initial centroids.

In the next section, some related work are briefly discussed. Then the proposed algorithm and its computational complexity are described in Section 3. Section 4 applies this clustering method to some standard data sets and reports its performance. Finally, conclusion of the paper is summarized in Section 5.

2. RELATED WORK

There exist several papers dedicated to agglomerative clustering [2, 3, 4]. Some algorithms [5–7] has attempted to perform agglomerative clustering on the graph representation of data like Chameleon [5] or graph degree linkage (GDL) [8]. Fränti et al. [9] proposed a fast PNN-based clustering using K-nearest neighbor graph with O(n log n) running time. Recently, Li et al. proposed a simple and accurate approach to hierarchical clustering [10], with a time complexity of $O(n^3)$, and a space complexity of $O(n^2)$, which is prohibitive when dealing with large data set. In [11], Chang et al. introduced a fast agglomerative clustering using information of k-nearest neighbors with time complexity $O(n^2)$. Zhang [12] proposed an agglomerative clustering based on Maximum Incremental Path Integral and claimed that extensive experimental comparisons showed that this algorithm outperforms the state-of-the-art clustering methods, without specifying its computational running time. Thus, to the best of our knowledge, the main limitation of existing agglomerative clustering methods is their high computational complexity.

Another drawback of many agglomerative clustering algorithms is their dependence on one ore more tuning parameters, which are often difficult to determine.

Besides agglomerative clustering, K-means [13] is among the most commonly used clustering algorithms, because its conceptual simplicity and its low computational complexity. However, K-means is sensitive to centroids initialization and may stuck in local optima.

To overcome this inherent drawback, several initialization methods for K-means have been developed, some of them are random methods [14,15], others methods like KKZ [16], principal components analysis (PCA) based partitioning, and Var-Part (variance partitioning)[17] are deterministic.

3. THE PROPOSED METHOD

This section firstly introduces ACM and then analyzes the computational time and space complexity.

Given a data set $X = \{X_1, X_2, \ldots, X_n\}$ in R^d, i.e., n points (vectors) each with d attributes, the goal of the clustering is to divide X into k exhaustive and mutually exclusive clusters $C = \{C_1, C_2, \ldots, C_k\}$, such that:

$$\bigcup_{1 \le i \le k} C_i = X \text{ , and } C_i \cap C_j = \emptyset \text{ for } 1 \le i, j \le k.$$

Clustering algorithms usually generate clusters by optimizing a criterion function. The most commonly used criterion function is the Sum of Squared Error (SSE) given by:

$$SSE = \sum_{i=1}^{k} \sum_{X_j \in C_i} \| X_j - m_i \|^2 \qquad (1)$$

where $\| . \|^2$ denotes the Euclidean norm and

$$m_i = (1/|C_i|) \sum_{X_j \in C_i} X_j$$

denotes the centroid of cluster C_i whose cardinality is $|C_i|$. The main idea of the proposed ACM method, is to choose k initial points (randomly or by using an appropriate initialization procedure) in the input data set X, which constitute the k initial singleton clusters. Then, the rest of X is scanned once, the distance between each non assigned point X_i and the nearest centroid m_j is compared with the minimum of the inter-cluster distances stored in a kxk D matrix. If it is lower, then X_i is assigned to cluster C_j, else the two clusters with closest centroids are merged together into one cluster and a singleton cluster is created with centroid X_i, seeking to minimize the SSE criterion. Then centroids and D matrix are updated. This process is repeated until all points in X are assigned. More details are presented in the following pseudo-code:

3.1 Pseudo-code of the proposed method

Input: A data set X whose cardinality is n and an integer k

Output: k cluster C_j

```
1 for i=1:k do
    $C_i \leftarrow \{X_i\}$
    $m_i \leftarrow X_i$
end for
2 $D \leftarrow (d(m_i,m_j))_{1 \le i \ne j \le k}$
  $mu \leftarrow \underset{i,j}{Min}(D)$ and $(a,b) \leftarrow Arg(\underset{i,j}{Min}(D))$
  $i \leftarrow k+1$
3 while $i \le n$ do
      $d_i \leftarrow \underset{j}{Min}(d(X_i,m_j))$
      $c \leftarrow Arg(\underset{j}{Min}\ d(X_i,m_j))$
    if $d_i < mu$ then
          $C_c \leftarrow C_c \cup \{X_i\}$
          $m_c \leftarrow (|C_c| m_c + X_i)/(|C_c|+1)$
          $D(c,:) \leftarrow (d(m_c,m_j))_{1 \le j \le k}$
          $D(:,c) \leftarrow D(c,:)'$
    else
          $C_a \leftarrow C_a \cup C_b$
          $m_a \leftarrow (|C_a| m_a + |C_b| m_b)/(|C_a| + |C_b|)$
          $C_b \leftarrow \{X_i\}$
          $m_b \leftarrow X_i$
          $D(a,:) \leftarrow (d(m_a,m_j))_{1 \le j \le k}$
          $D(:,a) \leftarrow D(a,:)'$
          $D(b,:) \leftarrow (d(m_b,m_j))_{1 \le j \le k}$
          $D(:,b) \leftarrow D(b,:)'$
    end if
    $i \leftarrow i+1$
    $mu \leftarrow \underset{h,j}{Min}(D)$ and $(a,b) \leftarrow Arg(\underset{h,j}{Min}(D))$
end while.
```

3.2 Time and Space Complexities

As shown in pseudo-code, at step 2, $O(k^2)$ operations are required to compute D matrix, and $O(k^2)$ space are required to store D matrix.

At step 3, the while loop is repeated n-k times, and updating D matrix, require only O(k) operations at each iteration. Thus, the overall running time complexity of ACM is O(nk) and its space complexity is $O(k^2)$.

4. EXPERIMENTAL RESULTS

To evaluate the proposed algorithm performance, experiments are done on some data sets from UCI Machine learning repository [18]. Clustering results of ACM were compared with those of K-means algorithm (KM) and ACM+KM which consists to initialize KM using the centroids found by ACM. Silhouette index [19,20] 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. A bigger average silhouette value indicates a better clustering result.

In order to obtain reproducible results, the compared methods are initialized with the same initial points (the first k points X_i in X) instead of random initialization and using the centroids found by ACM to initialize ACM+KM method. Average of computed silhouette values related to these experiments are reported in table 1 and some clustering results with their corresponding silhouette plots are shown in Fig. 1 to 4.

Table 1. **Average silhouette index values of clustering results related to the conducted experiments.**

Data set X	k	KM	ACM	ACM+KM
Iris	3	0.7786	0.7786	0.7786
Ruspini	4	0.6584	0.9086	0.9086
Aggregation	7	0.6543	0.6543	0.6709
Compound	6	0.6105	0.6309	0.6446
R15	15	0.6659	0.9361	0.9361
S1	15	0.7413	0.8761	0.8803
S2	15	0.6638	0.7840	0.8009
S3	15	0.5825	0.3663	0.6378
S4	15	0.5967	0.3886	0.6447
DIM32	16	0.5416	0.9962	0.9962
DIM64	16	0.4991	0.9985	0.9985
DIM128	16	0.3417	0.9991	0.9991
A1	20	0.6175	0.7751	0.7892
D31	31	0.5504	0.9218	0.9222

5. CONCLUSION

In this work, a simple and fast algorithm for agglomerative clustering was presented. It is easy to implement, without imposing any tuning parameter, except k, the number of clusters. Experiments conducted on some standard data sets demonstrate the effectiveness of the proposed approach and show that it could also be successfully used as an initialization method for the k-means algorithm.

Future research direction of this work would be focused on improving the accuracy of this method by: removing noisy data using an appropriate outlier detection procedure as a preprocessing step; starting with a suitable centroid initialization method; performing multiple random restarts, combined with a convenient optimization meta-heuristic procedure; using an appropriate similarity function instead of Euclidian distance, in order to find arbitrary shaped clusters.

6. ACKNOWLEDGMENTS

The authors sincerely thank the anonymous reviewers for their helpful comments.

7. REFERENCES

[1] 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.

[2] Franti, P., Virmajoki, O., Hautamaki, V.: Fast agglomerative clustering using a k-nearest neighbor graph. IEEE TPAMI 28(11) (2006) 1875–1881

[3] Cho, M., Lee, J., Lee, K.: Feature correspondence and deformable object matching via agglomerative correspondence clustering. In: ICCV. (2009)

[4] Sander, J., Ester, M., Kriegel, H., Xu, X.: Density-based clustering in spatial databases: The algorithm GDBSCAN and its applications. Data Mining and Knowledge Discovery 2(2)(1998) 169–194

[5] Karypis, G., Han, E., Kumar, V.: Chameleon: Hierarchical clustering using dynamic modeling. IEEE Computer 32(8) (1999) 68–75

[6] Zhao, D., Tang, X.: Cyclizing clusters via zeta function of a graph. In: NIPS. (2008)

[7] Felzenszwalb, P., Huttenlocher, D.: Efficient graph-based image segmentation. IJCV 59(2)

[8] Wei Zhang, Xiaogang Wang, Deli Zhao, Xiaoou Tang: Graph Degree Linkage: Agglomerative Clustering on a Directed Graph Computer Vision – ECCV 2012 Lecture Notes in Computer Science Volume 7572, 2012, pp 428-441

[9] Pasi Fränti, Olli Virmajoki and Ville Hautamäki:Fast PNN-based Clustering Using K-nearest Neighbor Graph. IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE,VOL. 28, NO. 11,NOVEMBER 2006

[10] Jianfu LI,, Jianshuang LI, Huaiqing HE:A Simple and Accurate Approach to Hierarchical Clustering. Journal of Computational Information Systems 7: 7 (2011) 2577-2584

[11] Chih-Tang Chang, Jim Z.C. Lai, M.D. Jeng: fast agglomerative clustering using information of k-nearest neighbors. Pattern Recognition 43 (2010) 3958–3968

[12] Wei Zhang , Deli Zhao, Xiaogang Wang:Agglomerative Clustering via Maximum Incremental Path Integral. Pattern Recognition 46(11) 3056-3065 (2013)

[13] S. Lloyd, “Least Squares Quantization in PCM,” IEEE Transactions on Information Theory, vol. 28, no. 2, pp. 129–136, 1982.

[14] P. S. Bradley and U. M. Fayyad, “Refining initial points for K-means clustering”, proceedings of the 15th International Conference on Machine Learning, (1998) July 24-27, Morgan Kaufmann, San Francisco, pp. 91-99.

[15] M. Al-Daoud and S. Roberts. New methods for the initialisation of clusters. Technical Report 94.34, School of Computer Studies,University of Leeds, 1994.

[16] I. Katsavounidis, C.-C. J. Kuo, and Z. Zhang. A new initialization technique for generalized Lloyd iteration. IEEE Signal Processing Letters,1(10):144–146, 1994.

[17] T. Su and J. G. Dy, “In Search of Deterministic Methods for Initializing K-Means and Gauss (2004) 167–181

[18] Merz C and Murphy P, UCI Repository of Machine Learning ftp://ftp.ics.uci.edu/pub/machine-Learning-databases Clustering datasets:http://cs.joensuu.fi/sipu/datasets/

[19] Kaufmann, L. and Rousseeuw, P.J. (1990) Finding Groups in Data. Wiley, New York.

[20] http://www.mathworks.com

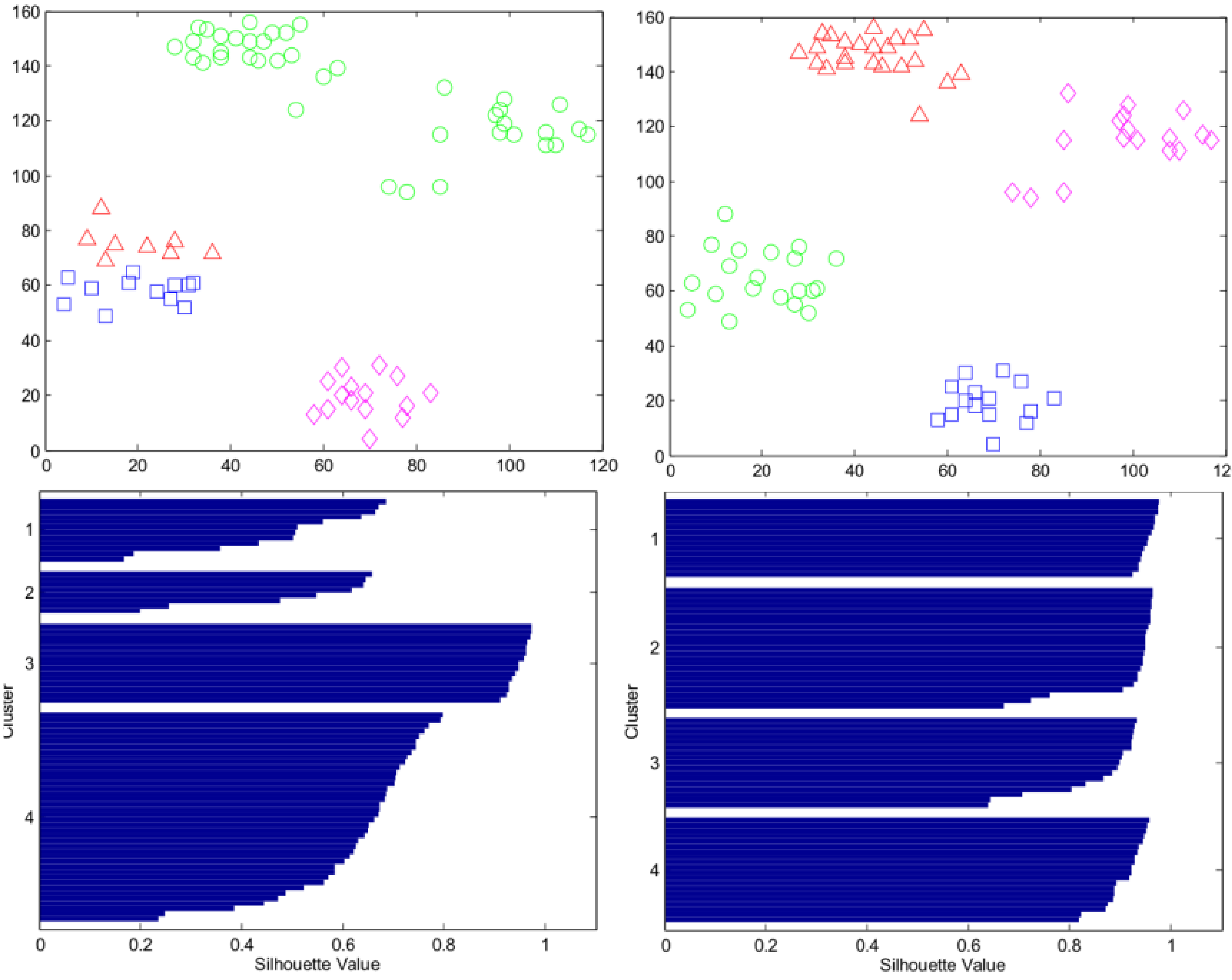

Fig 1: Clustering results of Ruspini data set using KM (on left) and ACM (on right)

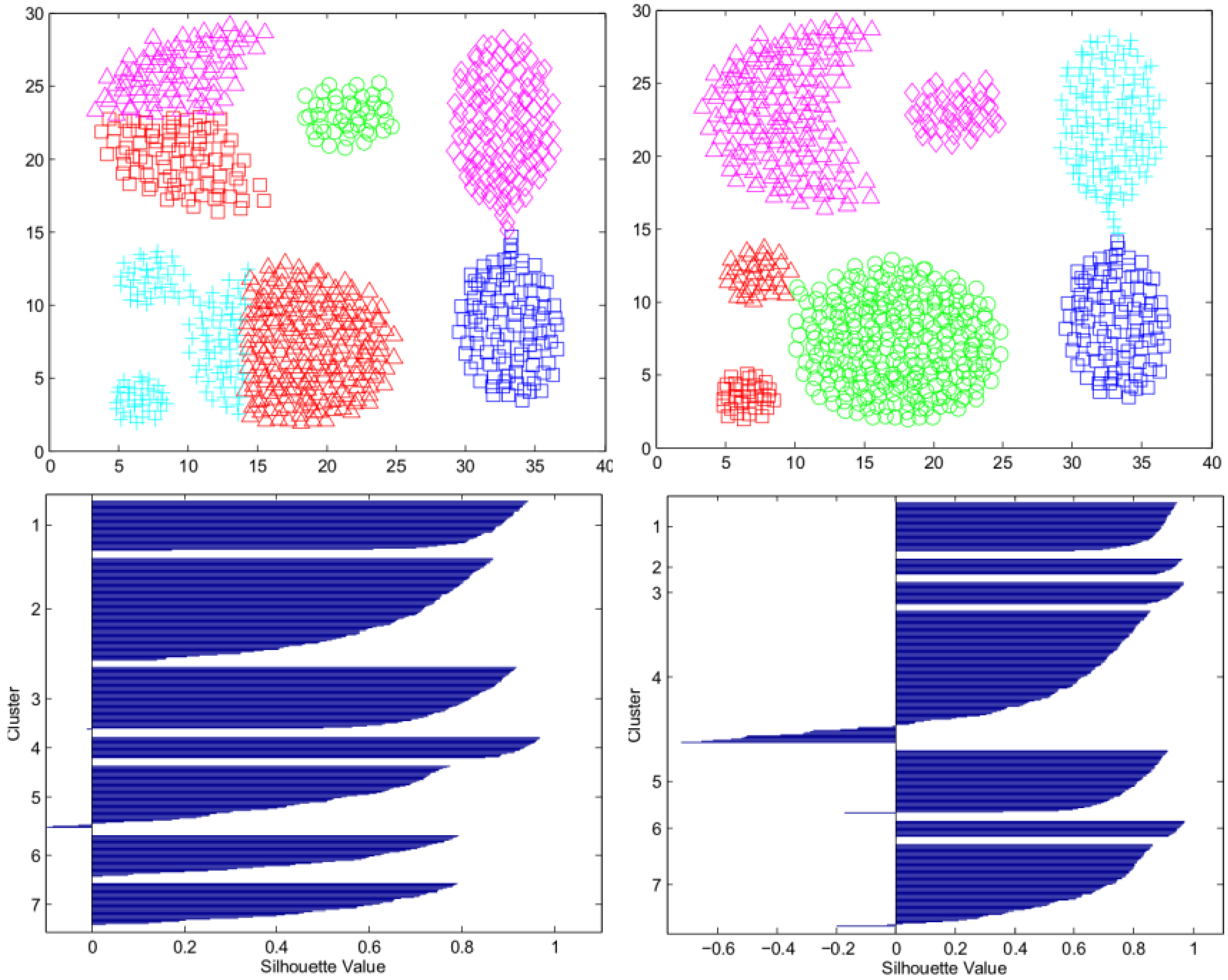

Fig 2: Clustering results of Aggregation data set using KM (on left) and ACM (on right)

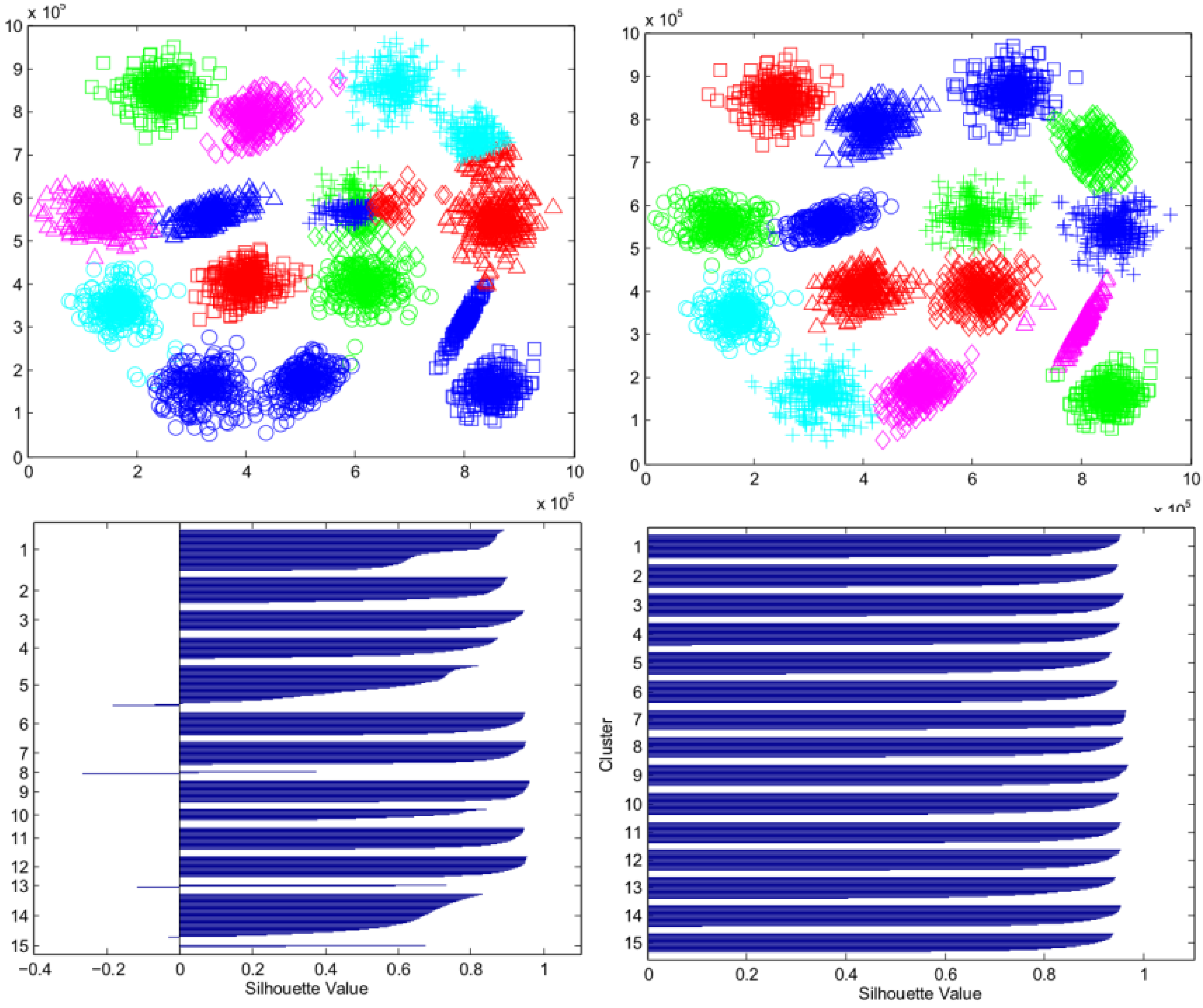

Fig 3: Clustering results of S1 data set using KM (on left) and ACM+KM (on right)

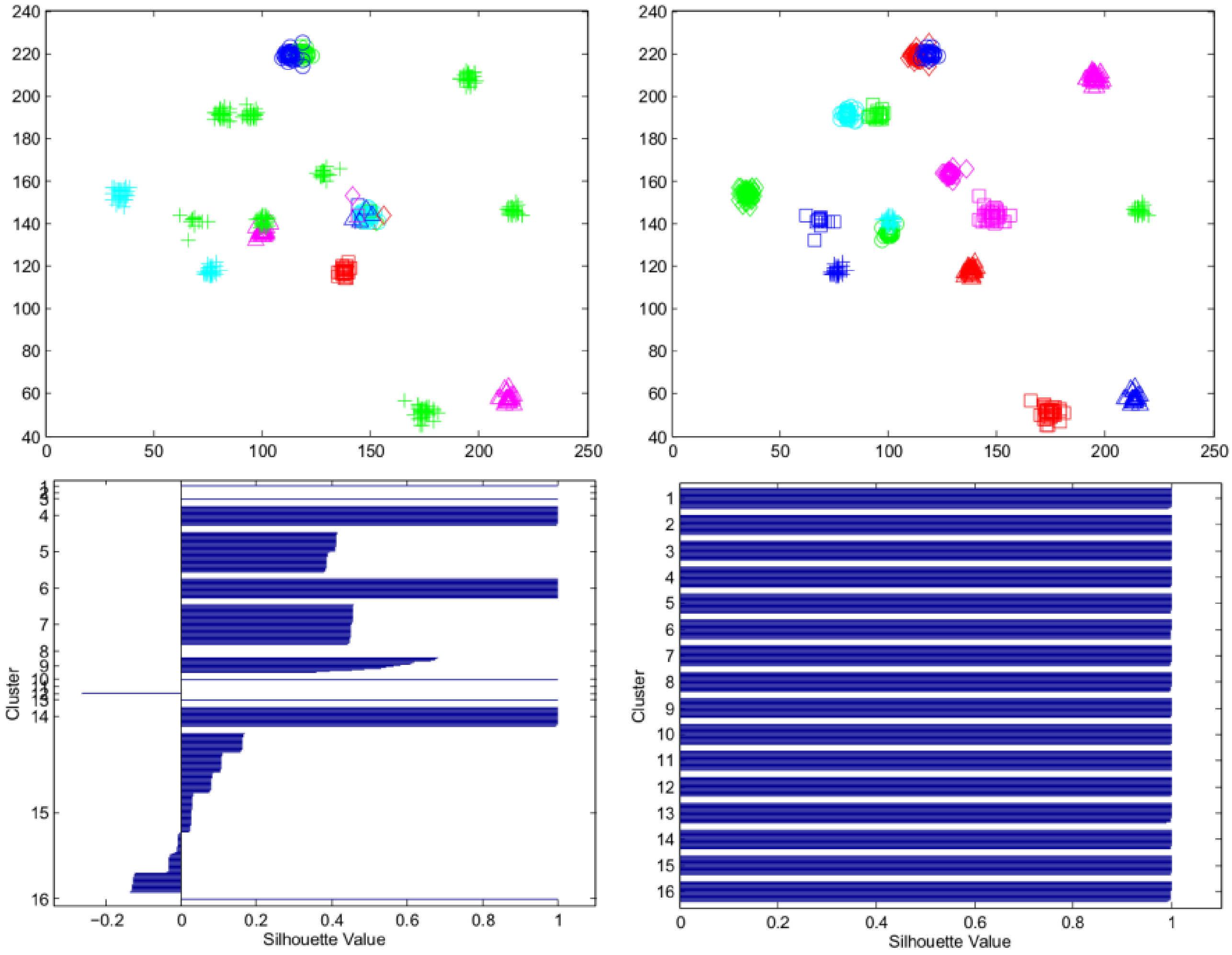

Fig 4: Clustering results of DIM128 data set using KM (on left) and ACM (on right)