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^5). 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 or 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:

\[ \bigcup_{C_i = X, \text{ and } C_i \cap C_j = \emptyset \text{ for } 1 \leq i, j \leq k. \]

1 \leq i \leq 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:

\[
SSE = \sum_{i=1}^{k} \sum_{X_j \in C_i} ||X_j - m_i||^2
\]

where \( || . ||^2 \) denotes the Euclidean norm and

\[
m_i = \frac{1}{|C_i|} \sum_{X_j \in C_i} X_j
\]
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 \( k \times k \) \( D \) 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 \( C_j \)

```plaintext
1 for i=1:k do
    \( C_i \leftarrow \{X_i\} \)
    \( m_i \leftarrow X_i \)
end for

2 \( D \leftarrow \{d(m_i, m_j)\}_{1 \leq i \neq j \leq k} \)

\( \mu \leftarrow \text{Min}(D) \) and \((a,b)\leftarrow \text{Arg}(\text{Min}(D))\)

\( \iota \leftarrow k+1 \)

3 while \( \iota < n \) do
    \( d_i \leftarrow \text{Min}(d(X_i, m_j)) \)
    \( j \)
    \( c \leftarrow \text{Arg}(\text{Min} d(X_i, m_j)) \)
end while.
```

### 3.2 Time and Space Complexities

As shown in pseudo-code, at step 2, \( O(k^3) \) 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 $X_1$ 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.

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

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


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)