HCuRMD: Hierarchical Clustering Using Relative Minimal Distances

Charalampos Goulas, Dimitrios Chondrogiannis, Theodoros Xenakis, Alexandros Xenakis, and Photis Nanopoulos

Computer Engineering and Informatics Department, University of Patrai, Greece
{goulasch,dchondrogiannis,xenakis,xenaki,phnano}@ceid.upatras.gr

Abstract. In recent years, the ever increasing production of huge amounts of data has led the research community into trying to find new machine learning techniques in order to gain insight and discover hidden structures and correlation among these data. Therefore, clustering has become one of the most widely used techniques for exploratory data analysis. In this sense, this paper is proposing a new approach in hierarchical clustering; named HCuRMD, which improves the overall complexity of the whole clustering process by using a more relative perspective in defining minimal distances among different objects.

Keywords: Artificial Intelligence, Machine Learning, Hierarchical Clustering, Relative Minimal Distances.

1 Introduction

On the rise of the 21st century, towards a new and emerging information society, the convergence of computers and telecommunication has led to a continuously, increasing production and storage of huge amounts of data for almost any field of human engagement. It has been estimated that Google handles over four million different queries in every minute [1], while almost seventeen terabytes of data is produced every day only by the users of Facebook and Twitter [2] and CERN produces even more than that in every hour [3].

Hence, if data is the recorded facts of human evolvement, information is the rules that govern them and while society is depending on and looking earnestly in finding them, artificial intelligence is the answer in discovering them. Under these circumstances, clustering, as an integral part of machine learning and artificial intelligence, has become one of the most widely used exploratory tools, with applications ranging from statistics, computer science and biology to education, social sciences and psychology.

In general, clustering is called the process of grouping a set of n physical or abstract objects $X = \{X_1, X_2, \dots, X_n\}$, based on their similarity, into $K \ll n$ groups $C_1, C_2, \dots, C_K$, so as to $C_k \subseteq X$ and $C_k \neq \emptyset \; \forall \; k = 1,2, \dots K$. However, a vital question remains: "how can researchers quantify the concept of similarity?" According to [4] similarity concepts cannot be easily determined accurately. And this is happening

R. Chbeir et al. (Eds.): AIAI 2015, IFIP AICT 458, pp. 440–447, 2015.
DOI: 10.1007/978-3-319-23868-5_32

because the whole process of clustering is not just a particular algorithm, but more a general problem of dispersions that needs a solution. Thus, researchers may use different models, in each of which can be proposed significantly different algorithms using alternate concepts of what constitutes a cluster and how this can be discovered effectively.

In this sense, there are partitioning methods, the main representative of which is the very well-known K-Means algorithm [5], that are trying to divide a search space into K sub regions while minimizing the inter-cluster distance, density-based algorithms such as DBSCAN [6] and OPTICS [7], that are searching for dense areas of a search space in order to create clusters, hierarchical or else connectivity-based clustering algorithms, that are using linkage criteria in order to bring together and merge different points or clusters, as well as more advanced techniques that are able to handle large spatial databases, as shown in [8] and [9].

2 Hierarchical Clustering

The clustering algorithms that belong to this corresponding category are seeking to build hierarchical, tree structures of nested clusters. Depending on the strategy they follow in order to achieve the desired result, these algorithms are divided into the following two sub-categories [10]:

- Agglomerative or else Bottom-Up.
- Divisive of else Top-Down.

On the one hand, the first ones start their operation considering each of the distinct objects as a separate cluster (bottom) followed by a gradual at each step merging of one pair of them that meets some certain criteria, till there is only one cluster (up). On the contrary, the second ones start from one unique cluster that contains all given objects (top) and gradually divide it into smaller and smaller clusters till each object corresponds to a separate cluster (down). However, despite this core strategic difference, in both cases, such algorithms are always using a linkage criterion in order either to decide which clusters will be combined or at which point an existing cluster will be divided.

2.1 Linkage Criteria

A linkage criterion determines the (dis)similarity between different clusters as a function of the distances of all pairs of objects within them. The most commonly used criteria are the Single-Linkage and the Complete-Linkage that can be described as follows.

Single-Linkage Criterion

The dissimilarity (distance) $D_{A,B}$ between two clusters A and B, with $a \in A$ and $b \in B$ the corresponding objects that belong to them, is equal to:

$$D_{A,B} = min\{d(a,b): a \in A, b \in B\} \quad (1)$$

, where $d(a,b)$ is a distance metric, such as Euclidean or Manhattan.

Note. The corresponding algorithm that uses this criterion merges at each step this pair of clusters that has the minimum value among all possible pairs (Fig. 1).

Complete Linkage Criterion

The dissimilarity $D_{A,B}$ between two clusters A and B is calculated as the maximum distance among all possible pairs of objects (one per cluster):

$$D_{A,B} = max\{d(\alpha,b): a \in A, b \in B\} \quad (2)$$

Note. The corresponding algorithm merges at each step this pair of clusters that has the minimum value in the above criterion among all possible pairs (Fig. 1).

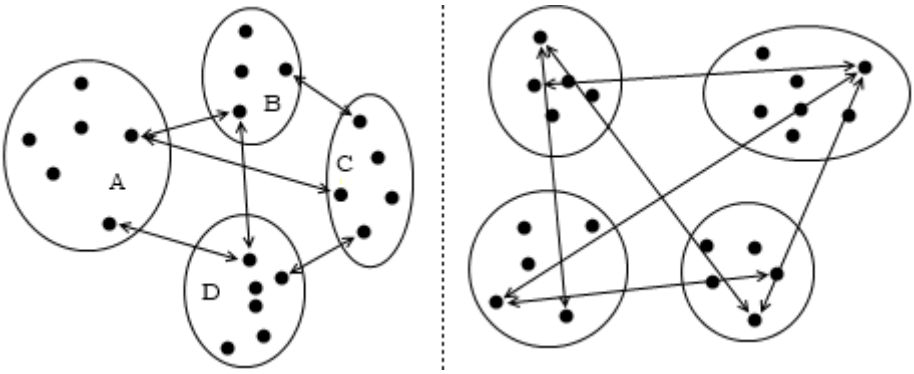

Fig. 1. Single-Linkage Criterion (left) vs. Complete Linkage Criterion (right)

Other Linkage-Criteria

Some other known, but not as frequently used linkage criteria are the followings [11]:

- **Mean or Average Linkage Criterion**

$$D_{A,B} = \frac{1}{|A||B|}\sum_{a\in A}\sum_{b\in B} d(a,b) \quad (3)$$

- **Minimum Energy Criterion**

$$D_{A,B} = \frac{2}{|A||B|}\sum_{i,j=1}^{|A||B|}\left\|a_i - b_j\right\|_2 - \frac{1}{|A|^2}\sum_{i,j=1}^{|A|}\left\|a_i - a_j\right\|_2 - \frac{1}{|B|^2}\sum_{i,j=1}^{|B|}\left\|b_i - b_j\right\|_2 \quad (4)$$

2.2 Time Complexity

On the one hand, the time complexity of most agglomerative clustering algorithms is $O(n^3)$, where n is the total number of the problem's objects, which makes them too slow in analyzing large datasets. On the other hand the time complexity of most divisive algorithms is $O(2^n)$, which in fact is even worse than the previous one. In general it has been shown that a hierarchical clustering algorithm can have $O\left(\mathrm{n}^2 \log_2{}^2 n\right)$ time complexity, independently of the clustering distance function [8, 14]. However, there have been found agglomerative algorithms, such as the single-linkage SLINK algorithm [12] and the complete-linkage CLINK [13] that are able to achieve an optimal time complexity of $O(n^2)$ for some special cases of problems.

3 Hierarchical Clustering Using Relative Minimal Distances

Given a set of N points $X = \{\overrightarrow{x_1}, \overrightarrow{x_2}, \dots, \overrightarrow{x_i} \dots, \overrightarrow{x_N}\}$ defined in $\mathcal{R}^p$ with Euclidian norm, where $\overrightarrow{x_i} = [x_{i1}, x_{i2}, \dots, x_{ip}]$ and p is the number of the problem's variables, the goal of the HCuRMD algorithm is to create $K << N$ clusters $= \{C_1, C_2, \dots, C_K\}$ so as to $C_k \subseteq X$, $C_k \neq \emptyset$ and $C_k \bigcap C_z = 0 \; \forall \; k \neq z \; and \; k, z = 1,2, \dots K$, while improving the overall time complexity of the provided solution.

In order to achieve the desired outcome, the algorithm firstly takes into account the following basic consideration:

- Each cluster C_k is represented by its center $\overrightarrow{m_k} = [m_{k1}, m_{k2}, \dots, m_{kp}]$, where m_{kj} is the mean of all observations x_{ij} of objects $\vec{x}_i \in C_k$ for the variable j:

$$m_{kj} = \frac{1}{|C_k|} \sum_{\vec{x}_i \in C_k, i=1}^{|C_k|} x_{ij} \tag{5}$$

Subsequently and according to the above, the algorithm is described in the sections that follow.

3.1 Rescaling the Data

Each variable represents a different feature of the given problem. The undesirable effect of different variables' measurement units can be eliminated by rescaling the data and essentially compressing the observations of all variables in the range [0, 1]:

$$sx_{ij} = \frac{x_{ij} - \min\limits_i(x_{ij})}{\max\limits_i(x_{ij}) - \min\limits_i(x_{ij})} \; \forall \; i = 1,2, \dots N \tag{6}$$

, where x_{ij} corresponds to the column with all observations of the variable j.

3.2 Calculating the Dissimilarity Matrix

At each iteration t a symmetric matrix D with dimensions $n_t x n_t$ is created, where n_t is the sum of the number of points that has not been used by the algorithm till this exact iteration plus the number of clusters that has been created, each of one is represented only by one point, its corresponding mean (Eq. 1). Therefore, each cell $d_{i,z}$ stores the square Euclidean distance between each possible pair of points $\overrightarrow{sx_i}$, $\overrightarrow{sx_z}$:

$$d_{i,z} = \sum_{j=1}^{p} \left(sx_{ij} - sx_{zj}\right)^2 \ \forall\, i, z = 1,2, \ldots n_t \tag{7}$$

3.3 Finding Nearest Neighbors

For each distinct point or cluster, the nearest neighbor is determined, regarding either the nearest point or the nearest cluster's center. More specifically, for each column z of the table D, the algorithm finds the cell $d_{i,z}$ with the smallest value greater than zero and stores the result in an auxiliary vector D', each cell d'_z of which holds the index i of the corresponding cell $d_{i,z}$ that is the nearest neighbor of the distinct point or cluster z:

$$d'_z = i, \text{so as to: } d_{i,z} = \min_i \{d_{i,z}\} \ \kappa\alpha\iota\ d_{i,z} > 0 \tag{8}$$

3.4 Pair Selection for Merging

Once the above discovery of the nearest neighbors of all existing points or clusters is completed, the next step is the selection of the most "appropriate" of them in order to be merged. However, what does the term "appropriate" mean? A pair of points is considered to be suitable for merging if the distance between its components is the "relatively" minimal one. But, what exactly does the term "relatively" mean? At this point, the algorithm proposes and uses a more relative perspective in defining minimal distances among different objects, called "Relative Minimal Distance" that can be described as follows.

Relative Distance

The distance between two objects (points or clusters), defined in any p-dimensional space, is not exclusively determined only by the distance metric that is used, but in fact it primarily depends on the surrounding environment in which the corresponding objects are located as well as by the nature of the observers themselves.

For example, the distance between two objects could be considered as a small one in a sparse region of a search space, but at the same time this exact distance could in fact be too large in a dense region of the same search space. On the other hand, the distance between the Earth and the Moon can be held as a large one by a casual

observer, but in fact it is considered to be a very small one by an astrophysicist, who is looking for another "Earth" in the chaotic area of space.

Relative Minimal Distance

In addition to the above description, the distance between two objects in a search space is considered to be minimal, if and only if each of these objects is nearest to the other one (1-1 nearest Neighbors), regardless of the actual distance between them that is measured by any of the existing distance metrics (Fig. 2).

Fig. 2. Pairs of Nearest Neighbors according to "Relative Minimal Distances" distance metric

Therefore, two points or clusters' centers i, z are selected for merging if and only if each of them in Table D' is carrying the other one as its nearest neighbor:

$$d'_z = i \; \kappa\alpha\iota \; d'_i = z \tag{9}$$

3.5 Repetition and Termination

The above process (from the step 2) is repeated until a termination criterion is achieved. A termination criterion could be the depth of the constructed tree that the algorithm would stop or the number of the clusters that the user would like to be created.

4 Discussion

Therefore and based on the above process a naturally extracted conclusion could be that using Relative Minimal Distances does not provide any advantage over existing techniques, as the ultimate shortest distances that are obtained in each iteration using some of the previously discussed linkage criteria are already included in the groups of the relative shortest distances that are obtained by the proposed methodology.

However, this observation, as shown clearly in Fig. 2, does not have absolute power. Even if the final result could in certain cases be the same, the automatic determination of multiple starting points, by the use of relative minimum distances, it provides certain advantages in this process of a hierarchical clustering, which can be summarized as follows:

1. Indirect parallelization of the process of clustering.
2. Better exploration and "reading" of space exploration and the structure of the problem.

3. Reduction of the time complexity in solving a given problem.
4. Ability to identify and address the endpoints.
5. Improving the management ability and analysis of sparse or multidimensional search spaces.

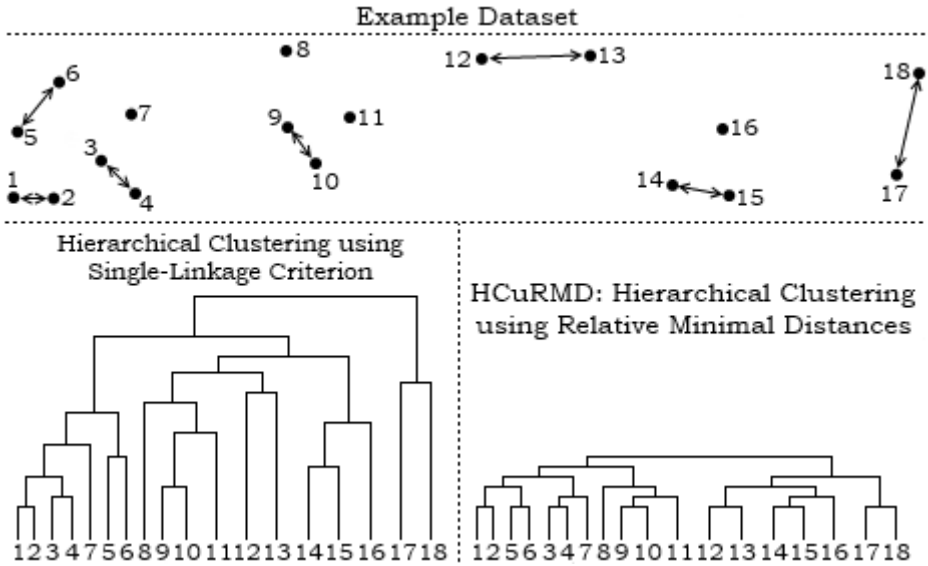

Fig. 3. Parallelization Process of Hierarchical Clustering using the "Relative Minimal Distance"

Subsequently and according to the above parallelization process, the proposed technique completes the operation in only 6 iterations, which is way faster of the corresponding process by using a single-linkage hierarchical clustering algorithm that needs 17 repetitions. In particular, the single-linkage clustering algorithm merges only the pair of clusters with the shortest distance between all possible pairs each time, a process that requires at least n-1 iterations, where n is the initial number of objects till the completion.

On the contrary, the proposed clustering algorithm by using the concept of "relative minimum distances" needs in best case scenario $\log_2 n$ repetitions, if all pairs merge into each repetition and $n-1$ repetitions in a single worst case scenario, where the $n-1$ distances between the corresponding objects show a gradual, incremental voltage (Fig. 4).

Fig. 4. Example of Worst Case Scenario in using "Relative Minimal Distances"

5 Conclusion and Evaluation

This paper presents a smart method for hierarchical clustering, named HCuRMD, which, unlike classical hierarchical clustering algorithms, does not take into consideration the actual Euclidean distances between all the pairs of objects, but instead considers only the 1-1 neighborhood graph. In particular, HCuRMD takes into account the mutual kNN graph for k=1 and drops the actual distances between the different objects. This approach drops unnecessary complexity and makes the algorithm terminate in fewer iterations ($\log_2 n$) in comparison to $n-1$ iterations that classical hierarchical clustering methods need.

References

1. Intel Corporation: What Happens in an Internet Minute?. http://www.intel.com
2. Zikopoulos, P., Eaton, C., de Ros, D., Deutch, T., Lapis, G.: Understanding Big Data, pp. 5–7. McGraw-Hill, USA (2012)
3. CERN: Computing. http://home.web.cern.ch/about/computing
4. Estivill-Castro, V.: Why so many clustering algorithms: A Position Paper. ACM SIGKDD Explorations Newsletter 4(1), 65–75 (2002)
5. Lloyd, P.S., Bell Telephone Laboratories.: Least square quantization in PCM. IEEE Transactions on Information Theory 28(2), 129–137 (1982); (First Written on 1957)
6. Ester, M., Kriegel, H.P., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD 1996), pp. 226–231 (1996)
7. Ankerst, M., Breunig, M.M., Kriegel, H.P., Sander, J.: OPTICS: Ordering Points to Identify the Clustering Structure. In: Proceedings of the 1999 ACM SIGMOD International Conference on Management of Data, pp. 49–60 (1999)
8. Nanopoulos, A., Theodoridis, Y., Manolopoulos, Y.: C2P: clustering based on closest pairs. In: Proceedings of the International Conference on Very Large Databases, pp. 331–340 (September 2001)
9. Corral, A., Manolopoulos, Y., Theodoridis, Y., Vassilakopoulos, M.: Algorithms for processing K-closest-pair queries in spatial databases. Data & Knowledge Engineering 49(1), 67–104 (2004)
10. Rokach, L., Oded, M.: Clustering methods. In: Data Mining and Knowledge Discovery Handbook, pp. 321–352. Springer, USA (2005)
11. SAS Institute Inc.: The CLUSTER Procedure: Clustering Methods, SAS/STAT 9.2 Users Guide, 2nd edn (2009). http://support.sas.com/documentation
12. Sibson, R.: SLINK: an optimally efficient algorithm for the single-link cluster method. The Computer Journal 16(1), 30–34 (1973)
13. Defays, D.: An efficient algorithm for a complete link method. The Computer Journal 20(4), 364–366 (1977)
14. Eppstein, D.: Fast Hierarchical Clustering and Other Applications of Dynamic Closest Pairs. In: Proc. Symposium on Discrete Algorithms, SODA 1998 (1998)