HAL open science

Dissimilarity Clustering by Hierarchical Multi-Level Refinement

Brieuc Conan-Guez, Fabrice Rossi

To cite this version:

Brieuc Conan-Guez, Fabrice Rossi. Dissimilarity Clustering by Hierarchical Multi-Level Refinement. 20-th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN 2012), Apr 2012, Bruges, Belgium. pp.483-488. hal-00692282

HAL Id: hal-00692282

https://hal.science/hal-00692282v1

Submitted on 29 Apr 2012

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Dissimilarity Clustering by Hierarchical Multi-Level Refinement

Brieuc Conan-Guez[1] and Fabrice Rossi[2]

1- LITA EA 3097, Université Paul Verlaine-Metz
Île du Saulcy 57045 Metz cedex 1, France

2- SAMM EA 4543, Université Paris 1 Panthéon-Sorbonne
90, rue de Tolbiac, 75634 Paris cedex 13, France

Abstract. We introduce in this paper a new way of optimizing the natural extension of the quantization error using in k-means clustering to dissimilarity data. The proposed method is based on hierarchical clustering analysis combined with multi-level heuristic refinement. The method is computationally efficient and achieves better quantization errors than the relational k-means.

1 Introduction

Non vector data arise in numerous real world applications for which complex object representations are needed, ranging from variable size strings to tree and graph structure observations. Such data can be analysed by means of dissimilarity measures: one works only on the square matrix of all pairwise dissimilarities between the observations. Numerous data analysis methods have been adapted or directly defined to handle dissimilarity matrices, from k-nearest neighbors in a supervised context to self organizing maps in the unsupervised one.

A traditional way of clustering proximity data is to apply the usual hierarchical clustering analysis (HCA) technique (see e.g. [1]). However, this elementary solution suffers from three potential difficulties. Firstly, classical linkage criteria for HCA such as single, average or complete, are not very satisfactory as they do not derive from a global quality criterion for the obtained partition. Secondly, naive implementations of the HCA scale in $O(N^3)$ (where N is the number of observations) which is unacceptable for large datasets. Thirdly, as a greedy technique, HCA leads to suboptimal partitions. We address in this paper those three limitations in order to derive an efficient alternative to prototype based dissimilarity clustering techniques such as the relational k-means [12]. Following [2], we introduce a linkage criterion which corresponds to the natural extension of the quantization error to dissimilarity data. Then we use the HCA algorithm from [8] to obtain a fast implementation. Finally, we apply multi-level refinement heuristics coming from graph clustering to improve the suboptimal partitions.

2 Hierarchical Clustering Analysis

2.1 Dissimilarity matrix and error measure

We consider the general case of a set Ω equipped with a dissimilarity measure. This is a map from $\Omega \times \Omega \to \mathbb{R}^+$, which is reflexive and symmetric, i.e.,

$d(i,i) = 0$ and $d(i,j) = d(j,i)$ for all $i, j \in \Omega$. We consider a partition of Ω in c distinct clusters: $\mathcal{P} = \{C_1, \ldots, C_c\}$. Denoting $S_{C_p,C_q} = \sum_{i \in C_p, j \in C_q} d(i,j)$, and $S_{C_p} = S_{C_p,C_p}$, the quality of a partition is measured thanks to the classical error measure [5]:

$$E(\mathcal{P}) = \sum_{C_k \in \mathcal{P}} \frac{1}{|C_k|} \sum_{i,j \in C_k} d(i,j) = \sum_{C_k \in \mathcal{P}} \frac{S_{C_k}}{|C_k|},$$

where $|C_k|$ is the size of cluster C_k. The normalization term $\frac{1}{|C_k|}$ can be seen as a way to favor partitions with clusters of intermediate sizes. It arises naturally in the case of vector data: indeed, as shown in e.g. [5], $E(\mathcal{P})$ is equal to twice the quantization error when the dissimilarity is the square of the Euclidean distance between observations.

Merging two clusters C_p and C_q of the partition $\mathcal{P}$ leads to a new partition $\mathcal{P}'$ with an error increased by:

$$\Delta E_{p,q} = \frac{S_{C_p \cup C_q}}{|C_p| + |C_q|} - \frac{S_{C_p}}{|C_p|} - \frac{S_{C_q}}{|C_q|}.$$

This increase can be used as a linkage criterion for a hierarchical clustering analysis (HCA). The derivation of this criterion generalizes Ward's derivation for the squared Euclidean distance (see [2]).

In a naive implementation of the HCA, each step of the algorithm proceeds by merging the pair of clusters, C_p and C_q, which leads to the smallest increase of the error E (that is the closest pair according to the linkage). If N denotes the size of Ω, the hierarchy is composed of a sequence of N partitions (from the finest to the coarsest). The algorithm complexity is $O(N^3)$, using the fact that computation of sums $S_{C_p \cup C_q}$ can be done incrementally, as $S_{C_p \cup C_q} = S_{C_p} + S_{C_q} + 2S_{C_p,C_q}$.

2.2 Fast Hierarchical Clustering Analysis

Since Lance and Williams introduced the generalized recurrence formula in 1967, many works have been devoted to produce efficient algorithms for agglomerative hierarchical clustering (see e.g. [7, 11]). In this work, we rely on a very recent algorithm proposed by D. Müllner [8]. Müllner's algorithm is a sophistication of an older algorithm proposed by Anderberg [1]. Even if the algorithm does not improve the worst case complexity of Anderberg's algorithm ($O(N^3)$), it is much more efficient in practical situations, as demonstrated by the author on various benchmarks [8]. Moreover this algorithm can handle general linkage criteria such as the one proposed in the previous section.

The efficiency of Müllner's algorithm can be explained succinctly as follows. As the search for the closest pair of clusters in term of linkage is the bottleneck of the naive implementation, Müllner's algorithm maintains a priority queue in which a nearest neighbor cluster candidate is stored for each cluster. The queue is sorted according to a lower bound of the distance between each cluster and its true nearest neighbor. At each step of the algorithm, the cluster of

highest priority (smallest lower bound) is dequeued from the structure, and the algorithm checks whether this cluster and its candidate are indeed the closest pair (the distance between the cluster and its candidate must be equal to the lower bound). In such a case, the merge is performed and the different data structures are updated. Otherwise, the true nearest neighbor is computed, the lower bound is replaced by the true distance to the nearest neighbor, and the cluster's candidate is again inserted in the priority queue. This approach leads in practice to a very efficient search for the closest pair of clusters. Additionally, it tends to delay as long as possible nearest neighbor computations. As these searches are postponed to latter steps in which the number of remaining clusters is smaller, the algorithm is much more efficient.

3 Multi-level refinement

3.1 Partition refinement

Once the hierarchy is set up, the analyst is free to choose a partition by cutting the dendrogram at a given level, using any adapted heuristics. However, the partitions obtained this way are frequently suboptimal. Indeed, as explained in e.g. [6], during the construction of the hierarchy, bad merging decisions during the early steps cannot be corrected in the later steps, which leads to a wrong clustering solution.

A quite simple approach to improve a given partition is to rely on a greedy refinement heuristic which performs a local search by moving objects from one cluster to another. The simplest approach is known as the *fast greedy heuristic*: the heuristic selects each object in turn; for a given object, the cluster switch that leads to the largest decrease in the error measure E is performed (all switches are considered). As long as moves which improve the partition quality exist, the process continues.

This heuristic has the advantage to be quite simple and can be implemented efficiently (calculations can usually be done incrementally). However, fast greedy has an important drawback, it can easily be trapped in local minima as only moves which decrease the error measure are allowed.

3.2 Multi-level refinement

In the context of hierarchical clustering, heuristics described in the previous section are known as single-level refinement approaches, as they operate on just one level of the hierarchy: the bottom level. Their main property is that they only move one object (a singleton) at a time. The Multi-level refinement (MLR) approach, on the opposite, operates on different levels of the hierarchy. For an intermediate level, each move corresponds to a displacement of several individuals (a cluster of the level in fact). This property allows the MLR to escape more easily from local minima than single-level heuristics. Indeed, for single-level heuristics, the displacement of a dense group of items is unlikely to happen, as it would imply many individual moves with a temporary large increase of the error

measure. For the MLR, on the opposite, the displacement of a group of items is done in just one operation, which avoids the increase of the error measure.

Operating on all the levels of dendrogram would be too costly as it would roughly multiply the cost of a refinement at a single level by $O(N)$. The MLR is therefore built upon a selection of levels. Given a reduction factor $0 < \alpha < 1$, the MLR considers only the initial trivial partition of singletons $\mathcal{P}_1$ with N clusters, and then a series of partitions $(\mathcal{P}_k)_{1 \leq k \leq K}$ whose sizes are given by a geometric progression based on α: $\mathcal{P}_2$ contains αN clusters, $\mathcal{P}_3$ $\alpha^2 N$ clusters, etc. $\mathcal{P}_{K-1}$ is the last level before the one chosen by the analyst (which corresponds to partition $\mathcal{P}_K$).

Given the series of partitions $(\mathcal{P}_k)_{1 \leq k \leq K}$, the MLR proceeds in a top-down way. As the partitions form a hierarchy, $\mathcal{P}_k$ can always be considered as a partition of the clusters of $\mathcal{P}_{k-p}$ for any $p > 0$. The main idea of MLR is to apply the fast greedy heuristic to $\mathcal{P}_K$ considered as a partition of $\mathcal{P}_k$ for k decreasing from $K-1$ to 1. More precisely, it first refines $\mathcal{P}_K$ considered as a partition of $\mathcal{P}_{K-1}$: this corresponds to moving entire sub-clusters of data points from one cluster of $\mathcal{P}_K$ to another one. Once this is done, the modifications of $\mathcal{P}_K$ are projected onto $\mathcal{P}_{K-2}$ and the process repeats. The final stage corresponds to applying fast greedy on the objects as in the single level approach.

Notice that moving a sub-cluster C of a given partition $\mathcal{P}_k$ from its current cluster C_p to another cluster C_q in the partition of interest $\mathcal{P}_K$ leads to an increase of the error measure:

$$\Delta E_{C:p \to q} = \frac{S_{C \cup C_q}}{|C| + |C_q|} - \frac{S_{C_q}}{|C_q|} - \frac{S_{C_p}}{|C_p|} + \frac{S_{C_p \setminus C}}{|C_p| - |C|}$$

Once again, sums $S_{C \cup C_q}$ and $S_{C_p \setminus C}$ can be updated incrementally at each move, leading to an efficient implementation.

3.3 Related works

Multi-level heuristics originate from graph clustering for which they give some of the best results (see [4] for the minimum cut partitioning problem, and [10] for the detection of communities in a network). MLR has also been considered for dissimilarity data in [6] as a way of improving HCA results. In this paper, the authors extract a k-nearest neighbour graphs from the dissimilarity matrix. They apply a standard HCA on the graph using a variant of the average linkage criterion. Then they apply a multi-level refinement approach to the hierarchical clustering using an error measure close to the quantity E used here. Our proposal differs in using the full dissimilarity matrix, in relying on the standard quantization error for dissimilarity data in all the phases of the algorithm and in leveraging Müllner's efficient HCA.

4 Experiments

The proposed method is tested on two classical dissimilarity data sets: the small size cat cortex database with 65 objects (see e.g., [3]) and the large size

Copenhagen chromosome database with 4200 objects (see e.g. [9]). Reference performances are provided by the relational k-means (RKM) in its standard (non naive) implementation described in [12]. In all cases, the RKM is started from a number of initial random configurations chosen so that both methods use approximately the same computational ressources (on the same computer and using the same implementation language, java).

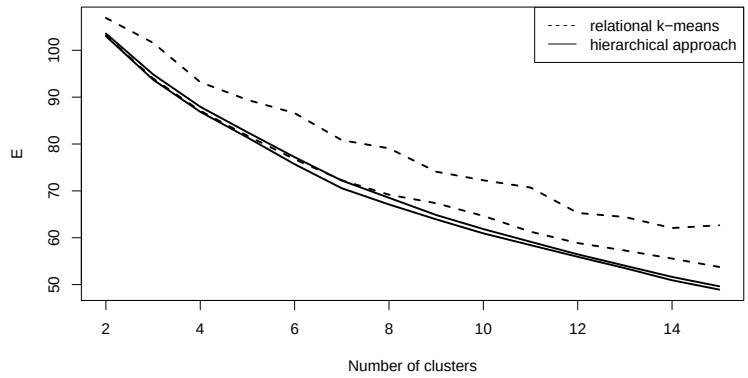

Fig. 1: Quantization error E as a function of the number of clusters for the cat cortex database (top solid line: standard HCA, bottom solid line: MLR, dashed lines: best and worst results of the RKM)

Results obtained on the cat cortex data set are summarized by Figure 1. For the relational k-means, the two dashed lines correspond to the best and the worst results obtained out of 20 random initial configurations. For the hierarchical approach, the top line corresponds to the quantization error after the hierarchical phase while the bottom line shows the error reached after the multi-level refinement. For the MLR, we set $\alpha = 0.75$. This value achieves a good balance between solution quality and MLR running time. The refinement does not bring much in this case (less than 2% of decrease of E), but it enables the hierarchical approach to beat the relational k-means in all cases. It should be noted that even with 100 times more initial configurations (a choice that increases significantly the running time of the method), the RKM reaches the same quality as the hierarchical method only for 2 to 5 clusters.

Similar results are obtained on the chromosome database (with only 10 random initial configurations due to a higher computational load for the RKM for this dataset), as shown on Figure 2. Both methods perform comparatively for a small number of clusters, while the hierarchical approach outperforms the relational k-means when the number of clusters increases.

5 Conclusion

As shown in the experiments, the proposed hierarchical approach gives very satisfactory results on real world datasets, especially when the number of clusters chosen by the analyst is high. Further works will focus on exploiting the full

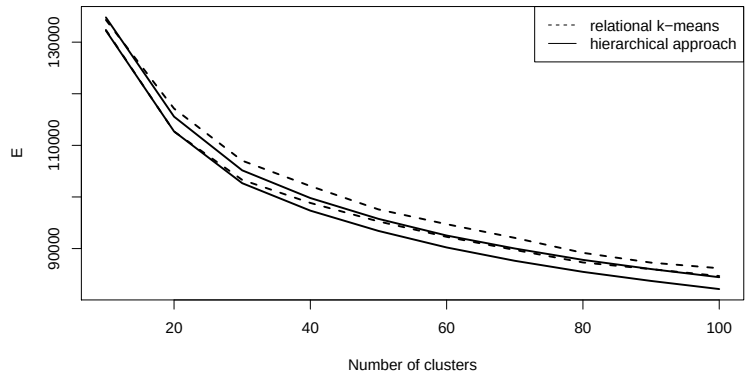

Fig. 2: Quantization error E as a function of the number of clusters for the chromosome database

hierarchy as we have only used here the method in a k-means like situation where only one clustering is considered.

References

[1] M. R. Anderberg. *Cluster analysis for applications.* Academic Press, New York, 1973.

[2] V. Batagelj. Generalized ward and related clustering problems ward clustering problem. In H. H. Bock, editor, *Classification and related methods of data analysis*, pages 67–74. North-Holland, Amsterdam, 1988.

[3] T. Graepel and K. Obermayer. A stochastic self-organizing map for proximity data. *Neural Computation*, 11(1):139–155, 1999.

[4] B. Hendrickson and R. Leland. A multilevel algorithm for partitioning graphs. In *Proceedings of the 1995 ACM/IEEE conference on Supercomputing (CDROM)*, Supercomputing '95, New York, NY, USA, 1995. ACM.

[5] T. Hofmann and J. M. Buhmann. Pairwise data clustering by deterministic annealing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(1):1–14, January 1997.

[6] G. Karypis, E. hong (Sam) Han, and V. Kumar. Multilevel refinement for hierarchical clustering. Technical report, University of Minnesota, 1999.

[7] F. Murtagh. *Multidimensional clustering algorithms.* Compstat lectures. Physica-Verlag, 1985.

[8] D. Müllner. Modern hierarchical, agglomerative clustering algorithms. *Lecture Notes in Computer Science*, 3918(1973):29, 2011.

[9] M. Neuhaus and H. Bunke. Edit distance-based kernel functions for structural pattern classification. *Pattern Recognition*, 39(10):1852–1863, 2006.

[10] A. Noack and R. Rotta. Multi-level algorithms for modularity clustering. In *Proceedings of the 8th International Symposium on Experimental Algorithms*, SEA '09, pages 257–268, Berlin, Heidelberg, 2009. Springer-Verlag.

[11] F. J. Rohlf. *Hierarchical clustering using minimum spanning tree*, volume 16. 1973.

[12] F. Rossi, A. Hasenfuss, and B. Hammer. Accelerating relational clustering algorithms with sparse prototype representation. In *Proceedings of the 6th International Workshop on Self-Organizing Maps (WSOM 07)*, Bielefeld (Germany), September 2007.