Robust Clustering by Aggregation and Intersection Methods

Ida Bifulco, Carmine Fedullo, Francesco Napolitano, Giancarlo Raiconi, and Roberto Tagliaferri

NeuRoNe Lab, DMI, University of Salerno, via Ponte don Melillo, 84084 Fisciano (SA) Italy
{ibifulco,fnapolitano,gianni,rtagliaferri}@unisa.it
http://www.neuronelab.dmi.unisa.it

Abstract. When dealing with multiple clustering solutions, the problem of extrapolating a small number of good different solutions becomes crucial. This problem is faced by the so called Meta Clustering [12], that produces clusters of clustering solutions. Often such groups, called meta-clusters, represent alternative ways of grouping the original data. The next step is to construct a clustering which represents a chosen meta-cluster. In this work, starting from a population of solutions, we build meta-clusters by hierarchical agglomerative approach with respect to an entropy-based similarity measure. The selection of the threshold value is controlled by the user through interactive visualizations. When the meta-cluster is selected, the representative clustering is constructed following two different consensus approaches. The process is illustrated through a synthetic dataset.

Keywords: consensus clustering, meta clustering, mds visualization, dendrogram visualization.

1 Introduction

In the last years many papers have been published in literature regarding the use of clustering techniques in the Knowledge Discovery and Data Mining field, for data analysis in many applications and scientific areas [3,4,5].

The main idea of cluster analysis is to group together similar patterns depending on the chosen features. The most used algorithms start from a random or arbitrary initial configuration and then evolve to a local minimum of the objective function. In complex problems (in many real cases) there are several minima and more than one can explain in a convincing manner the data distribution. In this case, we need, at least, to run many times the algorithms to choose more reasonable solutions. This is due to the intrinsic ill-posedness of clustering where the existence of a global solution cannot be assured and small perturbations of data due to noise can lead to very different solutions. Some discussions about this point can be found in [6,7,8].

As a consequence of these facts, in some cases researchers try to assess the stability and reliability of the obtained clusterings [11,10,9]. Stability means that

I. Lovrek, R.J. Howlett, and L.C. Jain (Eds.): KES 2008, Part III, LNAI 5179, pp. 732–739, 2008.

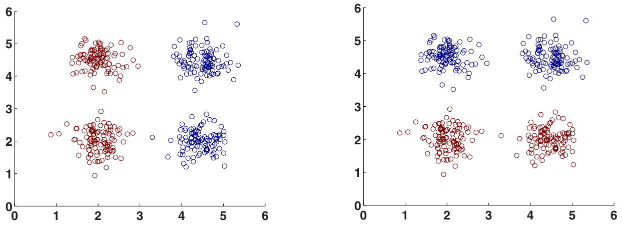

Fig. 1. Multiple minima example. The shown dataset has two equivalent and symmetric 2-clusters solutions.

slight perturbations on the inputs does not influence the obtained clusters too much. Reliability, instead is the tendency of a pattern to belong always to the same cluster with high probability or possibility.

Stability implies the concept of uniqueness of solution. For example, when using *k-means* the main problem is finding the parameter k assuring the best stability value. But k indicates also the zoom level we choose to analyze data. In some cases, once fixed k, we can have more than one stable solution. For example, suppose to have a mixture of four well separated Gaussians with the same number of patterns but different means and variance. It is obvious that in this case the best stability is obtained by solutions with four clusters. Let us suppose, instead, that we seek for a more rough grouping of data, for example with only two clusters. In this case there are two obvious different best solutions with respect to the distortion function that are equivalently reasonable. The two solutions cannot be merged together in a good way if we want to maintain k=2 or the complete dataset (Fig.1). The process of merging clustering solutions is called consensus and two approaches to this problem, one of which is introduced in this paper, are illustrated in Section 2. The other one was chosen for its ability to exclude points from the consensus, which is also a feature of our method. Meta clustering [12] is the process of clustering multiple solutions and is introduced in Section 1. Section 3 shows the application of such ideas to a synthetic dataset.

2 Meta Clustering

When dealing with multiple clustering solutions, the problem of extrapolating a small number of good different solutions becomes crucial. Grouping similar clustering solutions together is in turn a clustering problem. This process is called Meta Clustering [12]. Partly following the approach of [12], we divide the meta clustering process into 4 steps: 1) Generate many good different base-level clusterings (in our case local minima of the objective function generated by global optimization techniques [13]). 2) Measure the similarity between each couple of base clusterings. 3) Exploit the computed similarities to build a dendrogram and an MDS Euclidean embedding of the clustering solutions. 4) Use consensus algorithms to obtain a single solution from each meta-cluster.

Quality measures for the aggregations (such as accuracy, compactness etc.[12]) are used to decide the threshold of ”mergeability” of the clusterings: a high decrease of quality for an aggregated solution suggests that the base solutions should not be merged.

The use of interactive tools derived from our previous work ([15,14]) permits to give the user information he needs to make decisions on the best meta clustering solutions.

3 Consensus Clustering

Consensus clustering, also known in literature as clustering ensembles and clustering aggregation, is the process of extrapolating a single clustering solution from a collection, in such a way to maximize a measure of agreement. This problem is NP complete [16]. In [17] some methods are introduced for consensus proving a 3-approximation algorithm. On the other hand, [18] suggests that the complexity of approximation algorithms can be unjustified, given the comparable performances of heuristic approaches. They also indicate the combined use of unsampling techniques and approximation algorithms as the best compromise. In [19] three EM-like methods are illustrated showing comparable performance with other 11 algorithms. EM approach is also exploited in [20], but combining weak clusterings. In [21] consensus clustering was improved by adaptive weighted subsampling of data. In the following we shall consider two kinds of approach: the consensus method of [17], and a novel method that obtains a subpartition of data using the intersection between corresponding clusters belonging to all solutions that are included in a metacluster.

3.1 Clustering Aggregation

The first algorithm we use in this paper is the consensus ”Balls Algorithm” proposed in [17] for the correlation clustering problem. It first computes the matrix of pairwise distances between patterns. Then, it builds a graph whose vertices are the tuples of the dataset, and the edges are weighted by the distances. Aim of the algorithm is to find a set of vertices that are close to each other and far from other vertices. Given such a set, it can be considered a cluster and removed from the graph. The algorithm proceeds with the rest of the vertices. The algorithm is defined with an input parameter α that guarantees a constant approximation ratio.

3.2 Intersection Method

Let be $X = \{x_1, ..., x_N\},\ x_i \in \mathbb{R}^n$ a set of N data points and $\mathcal{C} = \{\mathcal{C}^1, ..., \mathcal{C}^m\}$ a set of partitions (clustering) of such data, we want to infer from these a new solution (clustering) in agreement with all the collected information. To achieve this we must first state a model for comparing different partitions. This problem is defined as *m-partition clustering* in [1] as follows: let $\Delta(A, B) = |(A \setminus B) \cup (B \setminus A)|$ for two sets (partitions) A and B and let Σ be a collection of m partitions

$\mathcal{C}^1, \mathcal{C}^2, \ldots, \mathcal{C}^m$ of Σ with each partition $\mathcal{C}^i = \{C_1^i, C_2^i, ..., C_k^i\}$ containing exactly the same number k of subsets. Let a *valid solution* be a sequence of m permutations $\sigma = (\sigma_1, \sigma_2, ..., \sigma_m)$ of $\{1, 2, ..., k\}$ that *aligns* the partitions. For any permutation ρ of $\{1, 2, ..., k\}$, $\rho(i)$ is the $i - th$ element of ρ for $1 \leqslant i \leqslant k$, we want to find σ that minimize :

$$f(\sigma) = \sum_{i=1}^{k} \sum_{1 \leq j < r \leq m} \Delta(C_j^{\sigma_j(i)}, C_r^{\sigma_r(i)}) \tag{1}$$

this is an NP-hard problem for $m > 2$ [2], so we have developed a heuristic approach based on a greedy strategy for it. The main steps performed by the procedure to achieve this objective are:

1. Clustering population ordering
2. Similarity matrix for cluster computation
3. Cluster ordering
4. Intersection of solutions

Clustering population ordering. Let Σ be a collection of m partitions $\{\mathcal{C}^1, \mathcal{C}^2, \ldots, \mathcal{C}^m\}$ with each partition $\mathcal{C}^i = \{C_1^i, C_2^i, ..., C_k^i\}$ containing exactly the same number k of subsets. The idea behind the developed algorithm is to view the sequencing problem of such collection as a graph optimization one. Consider a complete graph on which each node correspond to a clustering. The distance matrix has elements $d_{i,j} = 1 - S(i, j)$ where similarities S are computed by a symmetric version of the measure defined in [14]. In such terms finding a chain of neighbors can be viewed as finding a 'minimum path' connecting all nodes (a slight variation of the TSP problem). To approximately solve such last problem we use the following greedy strategy: at the first step the procedure selects from the initial set the clustering pair having the maximum similarity value and we define it *(Left,Right)*. Then, at each iteration, it is selected the nearest (from a similarity point of view) to *Left* and the nearest to *Right* and are added to the left and to the right of the current 'best clustering pair' and so on, until no clustering can be added to the solution.

Compute similarity matrix for cluster. Let be $\mathcal{C} = \{\mathcal{C}^1, ..., \mathcal{C}^m\}$, a sorted clustering set, and $C_j^i = \{C_1^i, ..., C_k^i\}$ for $i = 1, ..., m$, the k clusters of $i - th$ clustering. A similarity measure between clusters of a sorted clustering pair can be defined as follow:

$$\boldsymbol{sim}(C_j^i, C_l^{i+1}) = \frac{|C_j^i \bigcap C_l^{i+1}|}{|C_j^i|} \quad j, l = 1, ..., k \tag{2}$$

which summarizes the amount of common data between two clusters of lined up clustering solutions. This is a 'forward' measure because it takes into account only the similarity between clusters of successive clustering $(\mathcal{C}^i, \mathcal{C}^{i+1})$ for $i = 1, ..., m - 1$.

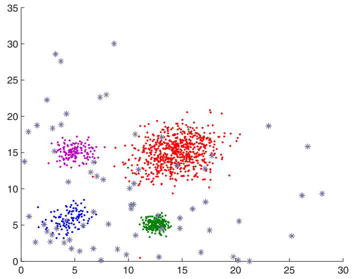

Fig. 2. Synthetic dataset with random noise

Cluster ordering. Let $\mathcal{C} = \{\mathcal{C}^1, ..., \mathcal{C}^m\}$ the set of clusterings sorted as described in the previous subsection, where each $\mathcal{C}^i = \{C_1^i, ..., C_k^i\} = \{set\ of\ k\ clusters\}$, and ***sim*** is a similarity matrix between clusters defined by (2). We can compute a greedy approximation for the cluster ordering problem as follows:

Step 1. Sort the clusters $\mathcal{C}^1$ such that[1]:
$|C_1^1| \geq |C_2^1| \geq ... \geq |C_k^1|$
Step 2. Set $\hat{C}_j^1 = C_j^1$ for $j = 1, ..., k$
Step 3. For each cluster in the current clustering and for each clustering in the solution set compute:

$$\hat{C}_j^{i+1} = \arg\max_{C_s^{i+1}}\{\boldsymbol{sim}(\hat{C}_j^i, C_s^{i+1})\} \quad C_s^{i+1} \neq \hat{C}_r^{i+1} \mid r < j \qquad (3)$$

note that the max value in (3) is computed taking into account only clusters not previously chosen.
Step 4. output the sequence of clusterings $\{\hat{\mathcal{C}}^1, ..., \hat{\mathcal{C}}^m\}$

Intersection of solutions. The final step of our procedure attempts to compute a consensus clustering among a set of ordered clusters by (3) as follows:

$$\tilde{C}_j^{intersect} = \{\hat{C}_j^1 \bigcap \hat{C}_j^2 \bigcap .... \bigcap \hat{C}_j^m\} \quad j = 1, ..., k \qquad (4)$$

4 Experimental Results

In order to test the proposed procedures we applied them to a synthetic dataset composed by a mixture of four Gaussians with different variances and additive noise (see Fig.2). We start by running the Optimization Algorithm (see [13]) to find local minima obtaining 36 different solutions. We build the hierarchical tree on the 36 solutions as shown in Fig.3, left, where leaves are labeled by numerals in ascending order of distortion value. The tree is built using the complete linkage method based on the similarity matrix as defined in [14]. We exploited interactive

[1] $|X|$ is the number of elements in set X.

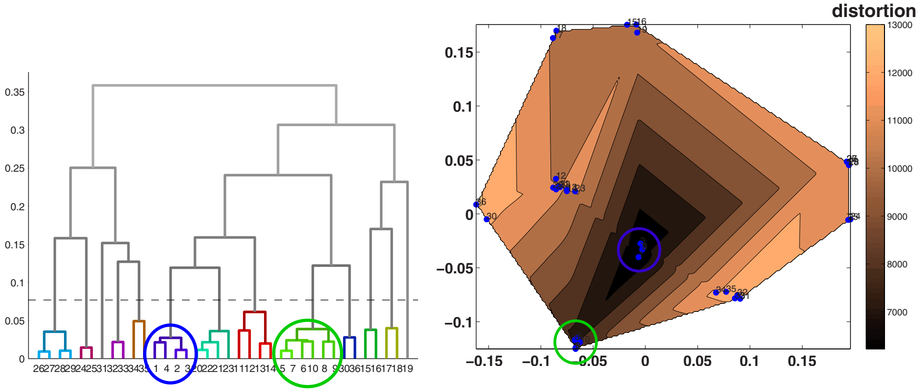

Fig. 3. Meta Clustering Dendrogram and MDS Clustering Map. Two subtrees have been chosen on the dendrogram and the solutions corresponding to their leaves are reported on the right map. They correspond to the two regions of minimum distorsion on the map.

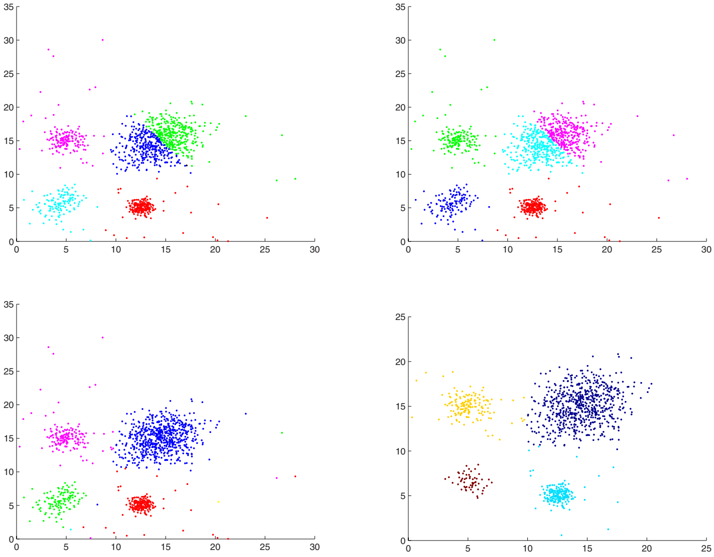

Fig. 4. Results of the two consensus approach performed on the two selected subtrees. First row, left to right: Ballclust on the blue subtrees (5 big clusters plus 2 singleton clusters), intersection method on the blue subtree (5 big clusters and 15 points discarded). Second row, left to right: Ballclust on the green subtree (4 big clusters plus 10 singleton clusters), intersection method on the green subtree (4 big clusters and 92 points discarded). It can be seen that the points discarded by intersection are mainly noise.

visualizations such as the hierarchical tree (see Fig.3, left), where the user can select any subtree; the MDS Clustering Map (see Fig.3, right), that put into evidence the solutions currently selected on the tree; the consensus solution performed on the leaves of the selected subtree (see Fig.4). Finally, some quality indicators are shown to the user, such as distortion values, number of clusters, number of points per cluster, number of points eliminated by the intersection method, mean dissimilarity between the solution and the leaves of the subtree, etc. Looking at figure 4 it is evident that both meta-clusters lead, using consensus techniques, to a final clustering that well explains the real data structure. The intersection method excludes more data from the final clustering, but results in a more robust solution.

5 Conclusions

In this paper we showed how Meta Clustering can be exploited to extract a small number of different and representative solutions, together with consensus algorithms. We used two different approaches to consensus clustering: the first method, known in literature, builds a new clustering by minimizing a disagreement function, while the second one, introduced in this paper, produces a subpartition by cluster intersection. All our tools are implemented in an interactive environment that simplifies the analysis of the results. Our future plans are to include the application of the most cited consensus algorithms in literature and the criteria to evaluate the quality of the proposed solutions in our interactive visualization tool.

References

1. Berman, P., DasGupta, B., Kao, M., Wang, J.: On constructing an optimal consensus clustering from multiple clusterings. Inf. Process. Lett. 104, (4) 137–145 (2007)
2. Gusfield, D.: Partition-distance: A problem and class of perfect graphs arising in clustering. Information Processing Letters 82, 159–164 (2002)
3. Amato, R., Ciaramella, A., Deniskina, N., et al.: A Multi-Step Approach to Time Series Analysis and Gene Expression Clustering. Bioinformatics 22, 589–596 (1995)
4. Jiang, D., Tang, C., Zhang, A.: Cluster Analysis for Gene Expression Data: A Survey. IEEE Transactions on Knowledge and Data Engineering 16, (11) 1370–1386 (2004)
5. Xui, R., Wunsch, D.: Survey of clustering algorithms. IEEE Transactions on Neural Networks 16(3), 645–678 (2005)
6. Hu, Y., Hu, Y.P.: Global optimization in clustering using hyperbolic cross points. Pattern Recognition 40(6), 1722–1733 (2007)
7. Agarwal, P.K., Mustafa, N.H.: k-means projective clustering. In: Proceedings of the Twenty-Third ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, pp. 155–165. ACM Press, New York (2004)
8. Kaukoranta, T., Franti, P., Nevalainen, O.: Reallocation of GLA codevectors for evading local minima. Electronics Letters 32(17), 1563–1564 (1996)

9. Valentini, G., Ruffino, F.: Characterization Of Lung Tumor Subtypes Through Gene Expression Cluster Validity Assessment. RAIRO-Inf. Theor. Appl. 40, 163–176 (2006)
10. Bertoni, A., Valentini, G.: Random projections for assessing gene expression cluster stability. In: Proceedings IEEE International Joint Conference on Neural Networks, vol. 1, pp. 149–154 (2005)
11. Kuncheva, L.I., Vetrov, D.P.: Evaluation of Stability of k-Means Cluster Ensembles with Respect to Random Initialization. PAMI 28(11), 1798–1808 (2006)
12. Caruana, R., Elhawary, M., Nguyen, N., Smith, C.: Meta Clustering. In: ICDM, pp. 107–118 (2006)
13. Bifulco, I., Murino, L., Napolitano, F., Raiconi, G., Tagliaferri, R.: Using Global Optimization to Explore Multiple Solutions of Clustering Problems. In: KES 2008 (2008)
14. Ciaramella, A., Cocozza, S., Iorio, F., Miele, G., Napolitano, F., Pinelli, M., Raiconi, G., Tagliaferri, R.: Interactive data analysis and clustering of genomic data. Neural Networks 21, 368–378 (2007)
15. Napolitano, F., Raiconi, G., Tagliaferri, R., Ciaramella, A., Staiano, A., Miele, A.: Clustering and visualization approaches for human cell cycle gene expression data analysis. International Journal Of Approximate Reasoning 47(1), 70–84 (2008)
16. Barthélemy, J.P., Leclerc, B.: The median procedure for partitions. In: Cox, I.J., Hansen, P., Julesz, B. (eds.) Partitioning Data Sets, American Mathematical Society, Providence, RI, pp. 3–34 (1995)
17. Gionis, A., Mannila, H., Tsaparas, P.: Clustering aggregation. ACM Trans. Knowl. Discov. Data 1(4), 1 (2007)
18. Bertolacci, M., Wirth, A.: Are approximation algorithms for consensus clustering worthwhile? In: 7th SIAM International Conference on Data Mining, pp. 437–442 (2007)
19. Nguyen, N., Caruana, R.: Consensus Clustering. In: Perner, P. (ed.) ICDM 2007. LNCS (LNAI), vol. 4597, pp. 607–612. Springer, Heidelberg (2007)
20. Topchy, A., Jain, A.K., Punch, W.: Clustering ensembles: models of consensus and weak partitions. IEEE Transactions on Pattern Analysis and Machine Intelligence 27(12), 1866–1881 (2005)
21. Topchy, A., Minaei-Bidgoli, B., Jain, A.K., Punch, W.F.: Adaptive clustering ensembles. Pattern Recognition. In: Proceedings of the 17th International Conference, vol. 1, pp. 272–275 (2004)