POLLACK PERIODICA
An International Journal for Engineering and Information Sciences
DOI: 10.1556/606.2019.14.1.2
Vol. 14, No. 1, pp. 15–24 (2019)
www.akademiai.com

FINDING THE REPRESENTATIVE IN A CLUSTER USING CORRELATION CLUSTERING

[1] Dávid NAGY[*], [2] Laszló ASZALÓS, [3] Tamás MIHÁLYDEÁK

[1,2,3] University of Debrecen, Faculty of Informatics
e-mail: [1]nagy.david@inf.unideb.hu, [2]aszalos.laszlo@inf.unideb.hu
[3]mihalydeak.tamas@inf.unideb.hu

Received 28 December 2017; accepted 14 June 2018

Abstract: Correlation clustering is a widely used technique in data mining. The clusters contain objects, which are typically similar to each other and different from objects from other groups. It can be an interesting task to find the member, which is the most similar to the others for each group. These objects can be called representatives. In this paper, a possible way to find these representatives are shown and software to test the method is also provided.

Keywords: Correlation clustering, Representative

1. Correlation clustering

Clustering is a widely used tool of unsupervised learning. Its task is to group objects in a way that the objects in one group (cluster) are similar, and the objects from different groups are dissimilar. This defines an equivalence relation. The similarity is usually based on the distance of the objects. However, sometimes only categorical data are given where distance is meaningless. For example: what is the distance between a cat and a dog? In this case, a tolerance relation is needed. Two objects can be treated as similar if this relation holds for these two objects. If the relation does not hold for two objects, then they are dissimilar. Naturally, this relation is reflexive because every object is similar to itself. It is also symmetric, which can be easy to see. The transitivity, however, does not necessarily hold. If a human and a mouse are taken, then due to their inner structure they are similar. This is the reason why mice are used in drug

[*] Corresponding Author

experiments. A human and a Paris doll are also similar due to their shape. This is why these dolls are used in show-windows. Although, a mouse and a doll are dissimilar (except that both are similar to the same object). Correlation clustering is a clustering technique, which is based on a tolerance relation [1], [2], [3]. The goal of correlation clustering is to find an equivalence relation, which is closest to the similarity (tolerance) relation. Let $\mathbf{V}$ be a set of objects and $\mathbf{T} \subset \mathbf{V} \times \mathbf{V}$ the tolerance relation. The result of correlation clustering is partition. This partition can be defined as a function: $\mathbf{p} = \mathbf{V} \to \{1, \cdots, n\}$. So it assigns for each object an integer number, which is its cluster IDentification number (ID). The objects **A** and **B** are in the same group if $\mathbf{p}(\mathbf{A}) = \mathbf{p}(\mathbf{B})$.

The following two cases can be treated as conflicts for two arbitrary objects **A** and **B**:

$$\begin{aligned} &\mathbf{ATB} \text{ holds,} && \text{but } \mathbf{p}(\mathbf{A}) \neq \mathbf{p}(\mathbf{B}), \\ &\mathbf{ATB} \text{ does not hold,} && \text{but } \mathbf{p}(\mathbf{A}) = \mathbf{p}(\mathbf{B}). \end{aligned} \tag{1}$$

The cost function *f* is the number of these disagreements. The value of the function *f* is the distance between the tolerance relation *T* and the equivalence relation defined by the partition. Solving a correlation clustering problem is equivalent to minimizing its cost function. The partition is called perfect if the cost function value is 0. It is easy to show that for an arbitrary tolerance relation, there is no necessarily perfect partition. *Fig. 1* shows a very simple example highlighting this issue.

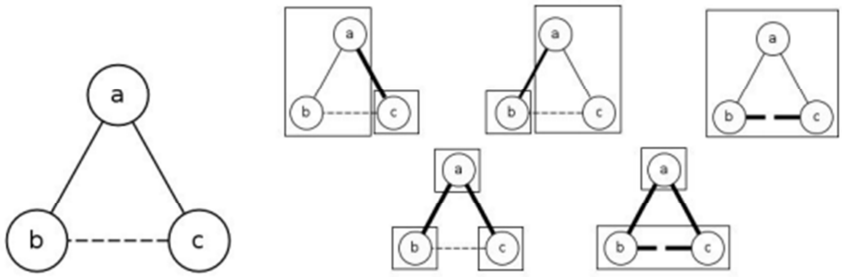

Fig. 1. Minimal frustrated similarity graph and its partitions

Take the relation on the left in *Fig. 1*. The dashed line denotes dissimilarity and the normal line similarity. The right parts of *Fig. 1* shows all the possible partitions of these objects, where rectangles indicate the clusters. The thick lines denote the pairs, which are counted in the cost function. In the upper row the value of the cost function is 1 (in each case), while in the two other cases it is 2 and 3, respectively.

Correlation clustering has many applications: image segmentation [4], identification of biologically relevant groups of genes [5], examination of social coalitions [6],

reduction of energy consumption [7], modeling physical processes [8], (soft) classification[9], [10], etc.

Despite its many applications, it has a disadvantage. It is a Nondeterministic Polynomial (NP) time complete problem, so it is very complicated to find the partition with minimal cost function value. The number of partitions also grows exponentially. It can be given by the Bell number [11]. In general - even in the case of some dozens of objects - the optimal partition cannot be determined in reasonable time. However, a quasi-optimal solution can be enough in practical cases. This can be achieved by using search algorithms. In this paper the authors used a genetic search algorithm [12]. This is a simple, well-known algorithm [13], [14], which can provide a rather good solution. Naturally, other search algorithms can be used.

2. Representative

The clusters gained from the correlation clustering contain the typically similar objects. In many cases, it can be interesting to find the member, which is the most similar to the other ones. This object can be treated as the representative because it can represent the whole group. If a decision about a certain group of objects is made, then in many cases it can be useful if only the representative member is considered. This can decrease the resource requirements because only one object for each cluster has to be considered.

Imagine that a product needs to be sold, for example a toy to a group of children. Almost every group of youngsters has at least one member whose decision has the most influence on the group's life. In this case one child needs to be found and convinced to buy the toy. The rest of the group will follow them afterwards.

Of course, if a group has more than one 'leader', then one from these possible leaders can be chosen.

If finding the representative needs to be formulated using mathematics, the following can be a possible way:

A member is called *representative* if it is similar to most of the members and different from the least of the members in the same cluster. So, for each member **U** four values have been stored:

- α - the number of elements that are similar to **U** and are in the same cluster;
- β - the number of elements that are different from **U** and are in the same cluster;
- γ - the number of elements that are similar to **U** and are in different clusters;
- δ - the number of elements that are different from **U** and are in different clusters.

In order to represent the idea of this paper, a proper similarity relation that can easily be visualized, is needed. The base of this relation is the Euclidean distance of the objects (d). Two thresholds were defined, one for similarity (S) and one for difference (D). The similarity relation (**T**) can be given the following way for each object **A, B**:

$$d(\mathbf{A},\mathbf{B}) = \begin{cases} +1, & \text{if} \quad d(\mathbf{A},\mathbf{B}) \le S, \\ -1, & \textit{if} \quad d(\mathbf{A},\mathbf{B}) > D, \\ 0 & \text{otherwise}. \end{cases} \tag{2}$$

Fig 2 shows a very simple example to the method. For the member **A** the four values are:

- α = 2 because there are two members (**B** and **C**) that are similar to **A** and are in the same cluster;
- β = 2 because there are two members (**F** and **E**) that are different from **A** and belong in the same cluster;
- γ = 2 because there are two members (**H** and **G**) that are similar to **A** and are in a different cluster;
- δ = 3 because there are three members (**J**, **K** and **L**) that are different from **A** and are in a different cluster.

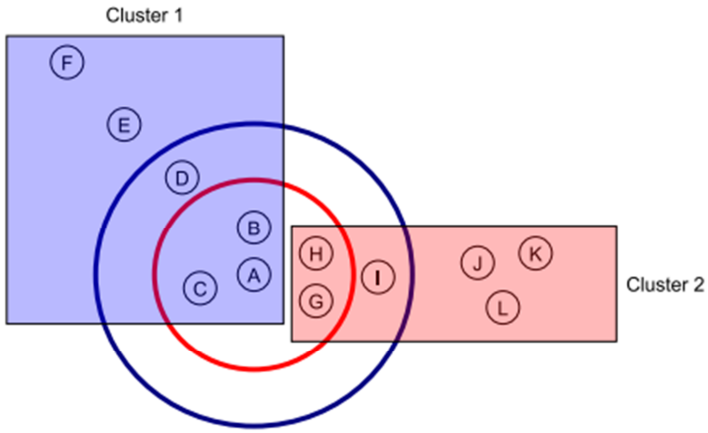

Fig. 2. α, β, γ, δ values of member **A**

The smaller circle denotes the similarity threshold and the greater one the difference threshold. There are two possible ways of defining the representative of a cluster *c*:

- If only the cluster *c* is considered, (first method);
- Every cluster is considered, (second method).

In case of the first one, a member can be considered as representative if the following fraction is maximal:

$$r_1 = \frac{\alpha^w - \beta^v}{\alpha + \beta + 1}, \quad \text{where} \quad v, w \in R, \quad v, w > 1, \quad w \geq v, \tag{3}$$

In case of the second method, a member can be considered as a representative if the following fraction is maximal:

$$r_2 = \frac{\alpha^w - \beta^v}{\alpha + \beta + u \cdot \gamma + 1}, \quad \text{where} \quad u, v, w \in R, \quad v, w > 1, \quad w \geq v, \tag{4}$$

If two arbitrary objects have the same r_2 value, then the δ value decides.

Of course, this method is only a possible way to define the representative members. Other similar methods can also be used.

The first method can be used when the members of the other groups do not matter. For example, let us assume that the objects are patients. Here, the similarity is based on having some common symptoms. If the patient, who is the most similar to the others, needs to be found, then the patients from the others groups are irrelevant. For instance, if the task is to find a new possible way to cure a certain disease that a group of patients has, then it can be useful to test it on the representative patient first. In this case, the other patients are not relevant because they have different symptoms.

The second method can be used when the members of the other groups matter. Let's assume that the objects are members of a political party. The similarity here can be based on the political view. Two politicians can be treated as similar if they share the same idea and different if they have different opinions. The leader of a party is expected to be similar to the others in the same party but different from the members of the other parties.

Another good but a little extreme example is if the objects are members of an organized crime family. Two gangsters are similar if they like each other and different if they do not. The boss of the family should be liked in the family but disliked in the other families. *Fig. 3* and *Fig. 4* show the difference between the two methods. In *Fig. 3* the first method was used. The member **A** was the representative of *cluster 1* because there are seven objects that are similar to **A**, and no such ones that are different from **A** (α = 7, β= 0, γ=2, δ = 2, v=2, w=2, u=1). So its r_1 value is maximal.

In *Fig. 4* the second method was used. Here, the member **F** was the representative of *cluster 1*. Its r_1 is less than that of member **A** because it has only 6 similar objects. However, the r_2 value is higher than that of member **A** because there are no objects that are similar to **F** and are in different clusters, while the member **A** has 2 objects (**I**, **J**) that are similar to it and are in *cluster 2* (α = 6, β= 0, γ=0, δ = 2, v=2, w=2, u=1).

3. The developed software

The authors of this article wrote software, which can help visualize the method. The software can be downloaded from [15].

The graphical user interface of the software can be seen in the following *Fig. 5*. First, the user gives the number of points, and then the points are generated in a two-dimensional interval, which is also given by the user. (These options can be given on the left panel of the user interface.) The base of the tolerance relation is the Euclidean distance of the objects as described in the previous section at (2).

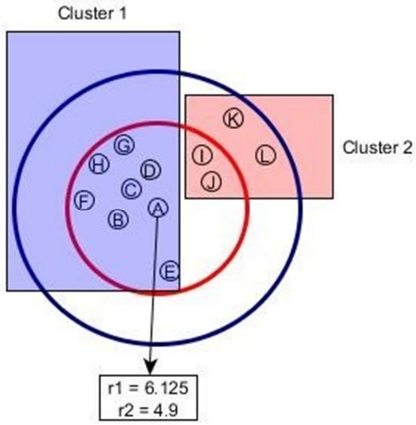

Fig. 3. The output of method 1

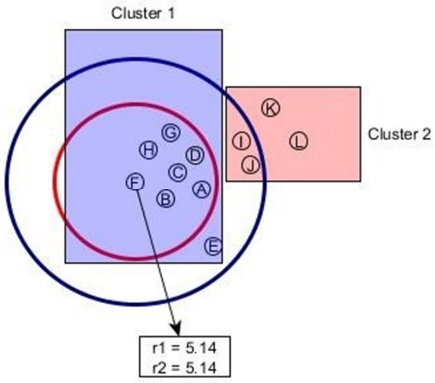

Fig. 4. The output of method 2

After generating the input points, the software finds a quasi-optimal partition using a genetic search algorithm. The pseudo-code of the algorithm, which is used in the software, can be seen in *Code 1* below. *Fig. 6* - *Fig. 8* show the output of the software for 50 points. The similarity threshold was set to 50 and the difference threshold to 90. The *v,w* weights were set to 2 and the *u* was set to 1, respectively.

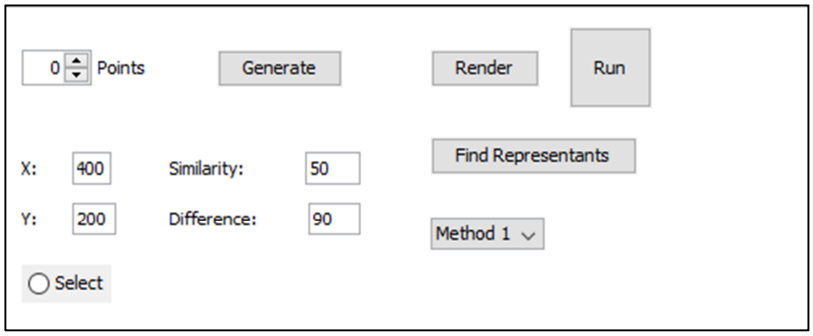

Fig. 5. Graphical user interface

Code 1

The pseudo-code of the algorithm

```
1: function FIND BEST PARTITION(N)
2:     population ← random_population
3:     while exit condition false do
4:         sort(population)
5:         for i ← 1,N,do
6:             new_population.add(population.get(i))
7:         end for
8:         p1 ← select_parents()
9:         p2 ← select_parents()
10:        children ← crossover(p1, p2)
11:        if small probability then
12:            mutation(children)
13:        end if
14:        new_population.add(children)
15:        population ← new_population
16:        max ← find_max(population)
17:    end while
18:    return max
19: end function
```

Fig. 6 shows the clusters generated by a genetic algorithm.

Fig. 7 presents the output of the first method. The representative is denoted by the plus sing. In almost every cluster the representatives are centroids. This was expected because the similarity was based on the Euclidean distance of the objects.

Fig. 8 presents the output of the second method. The most important difference can be observed in the cluster denoted by the star sign. In the second method, the representative is near the edge of the figure, so it is the farthest (most different) object in the cluster.

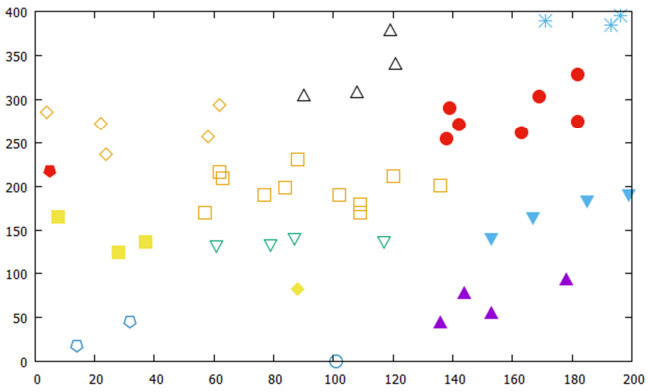

Fig. 6. The clusters of correlation clustering

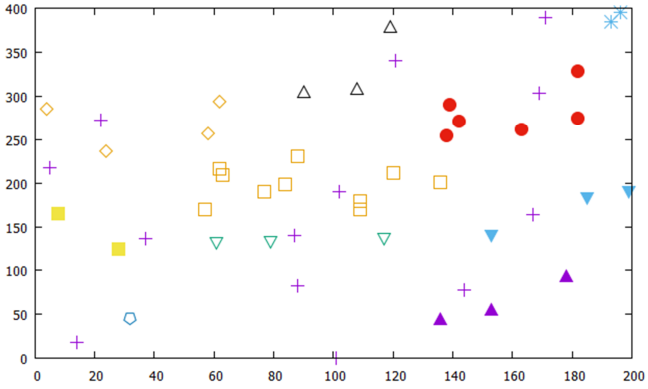

Fig. 7. The representative using method 1

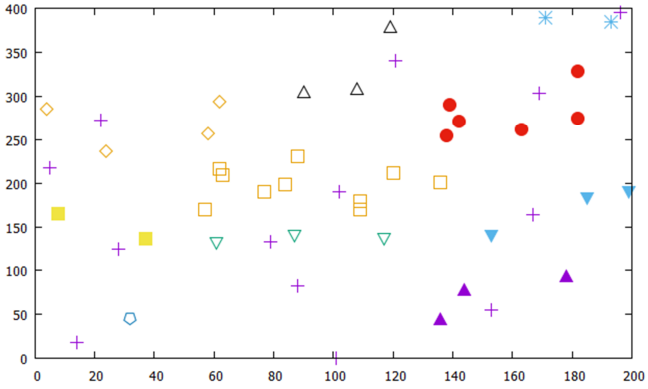

Fig. 8. The representative members using method 2

4. Conclusion

Correlation clustering is very effective method in many fields. In this paper the authors showed a possible way to define one object for each cluster, which is the most similar to the other objects in the same group. This member is called representative because it represents the others. The authors provided software, which can visualize the method. This program uses only random two-dimensional points. It can be interesting to test the method on real life data. In the paper, a genetic search algorithm was used. In the future, other algorithms could be implemented. It could be worth checking how the algorithms can affect the representatives.

Acknowledgements

This work was supported by the construction EFOP-3.6.3-VEKOP-16-2017-00002. The project was supported by the European Union, co-financed by the European Social Fund. This work was supported by the National Research, Development and Innovation Office of Hungary under Grant No. TÉT 16-1-2016-0193.

Open Access statement

References

[1] Bansal N., Blum A., Chawla S. Correlation clustering, *Machine Learning*, Vol. 56, No. 1-3, 2004, pp. 89–113.

[2] Becker H. A survey of correlation clustering, *COMS E6998: Advanced Topics in Computational Learning Theory*, 2005, pp. 1–10.

[3] Zimek A. Correlation clustering, *ACM SIGKDD Explorations Newsletter*, Vol. 11, No. 1, 2009, pp. 53–54.

[4] Kim S., Nowozin S., Kohli P., Yoo C. D. Higher-order correlation clustering for image segmentation, *Advances in Neural Information Processing Systems*, Vol. 24, 2011, pp. 1530–1538.

[5] Bhattacharya A., De R. K. Divisive correlation clustering algorithm (dcca) for grouping of genes: detecting varying patterns in expression profiles, B*ioinformatics*, Vol. 24, No. 11, 2008, pp. 1359–1366.

[6] Yang B., Cheung W. K., Liu J. Community mining from signed social networks, *IEEE Transactions on Knowledge and Data Engineering*, Vol. 19, No. 10, 2007, pp. 1333–1348.

[7] Chen Z., Yang S., Li L., Xie Z. A clustering approximation mechanism based on data spatial correlation in wireless sensor networks, *IEEE Wireless Telecommunications Symposium,* Tampa, FL, USA, 21-23 April 2010, pp. 1–7.

[8] Neda Z., Florian R., Ravasz M., Libál A., Györgyi G. Phase transition in an optimal clusterization model, *Physica A, Statistical Mechanics and its Applications*, Vol. 362, No. 2, 2006, pp. 357–368.
[9] Aszalos L., Mihálydeák T. Rough clustering generated by correlation clustering, In: Ciucci D., Inuiguchi M., Yao Y., Ślęzak D., Wang G. (Eds) *Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing*, Lecture Notes in Computer Science, Vol. 8170, Springer, Berlin, Heidelberg 2013, pp. 315–324.
[10] Aszalos L., Mihálydeák T. Rough classification based on correlation clustering, In: Miao D., Pedrycz W., Ślęzak D., Peters G., Hu Q., Wang R. (Eds) *Rough Sets and Knowledge Technology,* Lecture Notes in Computer Science, Vol. 8818. Springer, 2014, pp. 399–410.
[11] Aigner M. Enumeration via ballot numbers, *Discrete Mathematics*, Vol. 308, No. 12, 2008, pp. 2544–2563.
[12] Goldberg D. E., Holland J. H. Genetic algorithms and machine learning, *Machine Learning*, Vol. 3, No. 2, 1988, pp. 95–99.
[13] Kinczer T. Šulek P. The impact of genetic algorithm parameters on the optimization of hydro-thermal coordination*, Pollack Periodica*, Vol. 11, No. 2, 2016, pp. 113–123.
[14] Hatwagner M., Horvath A. Error handling techniques of genetic algorithms in parallel computing environment, *Pollack Periodica*, Vol. 3, No. 2, 2008, pp. 3–14.
[15] GitHub, https://github.com/lordimp88/representative (last visited 27 December 2017)