Technical notes on the `anticlust` package

Martin Papenberg

2019-02-27

This document explains the technical and algorithmical background of the `R` package `anticlust`. **It is still a work in progress.** The following topics are covered:

1. A formalization of the anticlustering problem
2. A description of the objective functions used to measure anticluster similarity
3. A documentation of the algorithms used to optimize the objective funtions

1 Problem formalization

A set of n d-dimensional data points $X = \{x_i\}$ $(i \in \{1, ..., n\})$ has to be partitioned into K clusters $C = \{c_k, k = 1, ..., K\}$, satisfying the following restrictions:

$$\bigcup_{k=1}^{K} c_k = X \tag{1}$$

$$S_k \cap S_j = \emptyset, \; \forall k, j \in \{1, ..., K\}, \; k \neq j \tag{2}$$

$$|c_k| = |c_j|, \; \forall k, j \in \{1, ..., K\} \tag{3}$$

Restriction (1) ensures that each element from the underlying set X is assigned to an anticluster; restriction (2) ensures that each element is assigned to only one anticluster; restriction (3) ensures that each anticluster contains the same number of elements. It follows that $|c_k| = \frac{n}{K} \; \forall k \in \{1, ..., K\}$. Restriction (3) is currently implemented for all methods in the `anticlust` package, but it is not an obligatory restriction for anticlustering in general.The objective is to select a partitioning that maximizes the similarity of the K anticlusters.

2 Objective functions

This section presents definitions of optimal anticluster similarity as assumed in the `anticlust` package.

2.1 The variance objective

Späth (1986) and Valev (1998) independently proposed to maximize the variance criterion used in k-means clustering as the objective in anticlustering. The variance criterion is given by sum of the squared errors between cluster centers (μ_k) and individual data points (Jain, 2010):

$$\sum_{k=1}^{K} \sum_{x_i \in c_k} ||x_i - \mu_k||^2 \tag{4}$$

The following plot graphically illustrates efforts to maximize and to minimize the variance criterion in a 2-dimensional feature space for three (anti)clusters, respectively. The partitions employ restrictions (1) - (3), including the restriction of equal (anti)cluster sizes. Optimizing the variance objective is a computationally difficult problem that is usually tackled using heuristic methods (Jain, 2010; Späth, 1986; Valev, 1998).

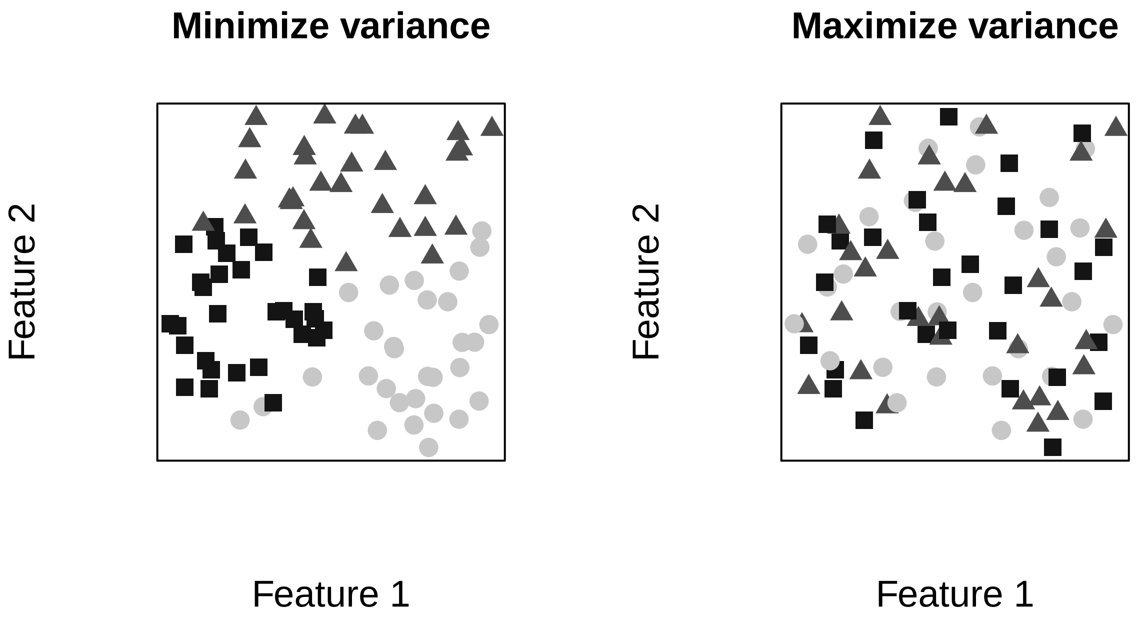

Figure 1: Attempts to minimize and maximize the variance objective, leading to clustering and anticlustering partitions, respectively.

2.2 The distance objective

In addition to the variance criterion, the `anticlust` package introduces another clustering objective to the anticlustering application. The objective has been developed in the problem domain of cluster editing and is based on a measure of the pairwise dissimilarities of data point (Böcker & Baumbach, 2013; Rahmann et al., 2007).[1] In weighted cluster editing, the optimal objective is found when the sum of within-cluster dissimilarities is minimized; for the anticlustering application, the objective is maximized instead.

To formalize the cluster editing objective, we use variables x_{ij} to encode whether two data points x_i and x_j belong to the same anticluster c_k:

$$x_{ij} = \begin{cases} 1 & \text{if } x_i \in c_k \wedge x_j \in c_k \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

Assume that d_{ij} represents a measure of the dissimilarity between two data points x_i and x_j, for example given as the euclidean distance. The cluster editing distance objective is then given as follows (Grötschel & Wakabayashi, 1989; Miyauchi & Sukegawa, 2015):

$$\sum_{1 \leq i < j \leq n} d_{ij} \, x_{ij} \tag{6}$$

The cluster editing objective is given as the sum of pairwise dissimilarities within the same cluster. I refer to this objective function as the "distance objective" as opposed to the "variance objective" in (4).

Maximizing the distance objective corresponds to minimizing the average linkage distance between anticlusters. In hierarchical cluster algorithms, the average linkage distance is a quantification of the similarity of two

[1] Cluster editing has also been studied under different names such as correlation clustering (Bansal, Blum, & Chawla, 2004), clique partition problem (Grötschel & Wakabayashi, 1989), and transitivity clustering (Wittkop et al., 2010).

clusters (Bacon, 2001; Guha, Rastogi, & Shim, 1998). To appreciate the correspondence of the distance objective and the average linkage method, consider the total sum of all pairwise dissimilarities. The total sum can be partitioned into within-cluster and between-cluster dissimilarities:

$$\sum_{1 \leq i < j \leq n} d_{ij} = \sum_{1 \leq i < j \leq n} d_{ij}\, x_{ij} + \sum_{1 \leq i < j \leq n} d_{ij}\, (1 - x_{ij}) \tag{7}$$

The total sum of distances is not influenced by the concrete partitioning x_{ij}; hence, the following optimizations lead to the same partitions, i.e., assignments of elements to anticlusters:

$$\text{Maximize} \sum_{1 \leq i < j \leq n} d_{ij}\, x_{ij}$$

$$\text{Minimize} \sum_{1 \leq i < j \leq n} d_{ij}\, (1 - x_{ij})$$

In the special case of $K = 2$ where we have two partitions A and B, the sum of the between-cluster distances can be formulated as follows:

$$\sum_{i \in A} \sum_{j \in B} d_{ij} \tag{8}$$

This formulation is very close to the average linkage objective that however incorporates the cardinalities of the sets A and B (Guha et al., 1998):

$$\frac{1}{|A|\,|B|} \sum_{i \in A} \sum_{j \in B} d_{ij} \tag{9}$$

Given restriction (3) for the anticlustering problem, the partitions A and B are of equal size; therefore, (8) gives the same information with regard to the similarity of clusters as (9). Hence, the cluster editing objective is a generalization of the average linkage measure on more than two clusters; minimizing the distance criterion minimizes cluster dissimilarity as measured by the average linkage method.

3 Algorithmic approaches

Finding optimal data partitions usually corresponds to NP-complete problems (Arabie & Hubert, 1996; Bansal et al., 2004; Jain, 2010). For NP-complete problems, it is often infeasible to find the optimal objective, especially when n is large. To find the optimal solution for moderately large instances, `anticlust` employs integer linear programming. To process larger problem instances, `anticlust` uses heuristic methods based on repeated random sampling.

3.1 NP-completeness

In the following, I show that anticlustering using the distance criterion is NP-complete. First, distance anticlustering is in NP because the distance objective can be computed in polynomial time for a given partitioning; the summation of all distance values d_{ij} is in $O(n^2)$.

Second, I show that if an efficient algorithm exists to solve distance anticlustering in polynomial time, it is also possible solve the NP-complete balanced number partitioning problem in polynomial time (Mertens, 2001). In the number partitioning problem, we have a list of positive integers $a_1, a_2, ..., a_n$ and try to find a subset $A \subset \{1, ..., n\}$ that minimizes the partition difference

$$E(A) = \left| \sum_{i \in A} a_i - \sum_{j \notin A} a_j \right| \tag{10}$$

In the balanced version of number partitioning, we may impose the restriction of $|A| = \frac{n}{2}$ – assuming that n is even – corresponding to restriction (3) of equal cluster sizes in anticlustering (Mertens, 2001).

To convert the number partitioning formulation into a formulation of distance anticlustering, we define d_{ij} as the absolute difference representing the dissimilarity of two numbers:

$$d_{ij} := |a_i - a_j| \tag{11}$$

We thus obtain

$$E(A) = \sum_{i \in A} \sum_{j \notin A} d_{ij} \tag{12}$$

Using variables $x_{ij} \in \{0, 1\}$ to represent whether two numbers are both either in A or not, i.e.,

$$x_{ij} = \begin{cases} 1 & \text{if } (x_i \in A \land x_j \in A) \lor (x_i \notin A \land x_j \notin A) \\ 0 & \text{otherwise} \end{cases}$$

we obtain $E(A)$ as the distance anticlustering objective:

$$E(A) = \sum_{1 \leq i < j \leq n} d_{ij} \, x_{ij} \tag{13}$$

Hence, balanced number partitioning is a special case of distance anticlustering where

a) $K = 2$
b) each element is described by exactly one integer
c) d_{ij} is the absolute difference

If a polynomial-time algorithm exists that solves distance anticlustering, we can therefore solve the NP-complete balanced number partitioning in polynomial time. Hence, distance anticlustering is NP-complete.

3.2 Integer linear programming

TODO:rework this section (how to incorporate graph formulation with the rest? Problem: x_ij = part of same clique AND edge connects i and j, and this is the same in the ILP formulation)

Despite the NP-complete nature of cluster editing, integer linear programming (ILP) has been successfully used to find optimal solutions even for relatively large problem instances (Böcker, Briesemeister, & Klau, 2011; L. H. N. Lorena, Quiles, Carvalho, & Lorena, 2018). Integer linear programming identifies the values of *decision variables* that optimize a linear objective function.

For the ILP formulation to distance anticlustering, we represent the problem input as an undirected complete graph $G = (V, E)$ (Grötschel & Wakabayashi, 1989; cf. Schaeffer, 2007). Each vertex $v \in V$ represents an element from the input data that has to be assigned to an anticluster. Edges are unordered pairs $\{i, j\} \in E$ of vertices, representing the relationships between elements. The short form ij will be used to refer to edges. A cost function $w : E \to \mathbb{R}$ assigns a weight to each edge. In distance anticlustering, edge weights is given by the distance between the two elements connected by the edge, i.e., $w(ij) = d_{ij}$.

In distance anticlustering, we optimize the linear objective in (6) by finding the best possible constellation of the binary decision variables $x_{ij} \in \{0, 1\}$ encoding whether two elements x_i and x_j are part of the same anticluster.

To solve anticlustering, a subgraph $G' = (V, E')$ is selected maximizing the distance objective in (6). G' must consist of disjoint cliques representing the anticlusters. In a clique, all vertices are connected by an edge, but there are no edges connecting different cliques. One has $ij \in E' \Leftrightarrow x_{ij} = 1$ and $ij \notin E' \Leftrightarrow x_{ij} = 0$ where x_{ij} indicate whether two vertices are connected by an edge, see (5).

Additionally, an ILP can employ constraints on the decision variables to satisfy external restrictions, implemented as mathematical inequalities. In the case of anticlustering, we have to implement inequalities ensuring that the anticlustering conditions defined in (1) - (3) are met. To this end, the `anticlust` package employs an ILP on the basis of formulations proposed by Grötschel and Wakabayashi (1989):

$$\text{Maximize} \sum_{1 \leq i < j \leq n} w(ij)\, x_{ij} \tag{14}$$

$$-x_{ij} + x_{ik} + x_{jk} \leq 1, \qquad \forall\, 1 \leq i < j < k \leq n, \tag{15}$$

$$x_{ij} - x_{ik} + x_{jk} \leq 1, \qquad \forall\, 1 \leq i < j < k \leq n, \tag{16}$$

$$x_{ij} + x_{ik} - x_{jk} \leq 1, \qquad \forall\, 1 \leq i < j < k \leq n, \tag{17}$$

$$\sum_{1 \leq i < j \leq n} x_{ij} + \sum_{1 \leq k < i \leq n} x_{ki} = \frac{n}{K} - 1, \qquad \forall i \in \{1, ..., n\} \tag{18}$$

$$x_{ij} \in \{0, 1\},\ \forall\, 1 \leq i < j \leq n \tag{19}$$

The inequalities (15) - (17) are called triangular constraints and were developed by Grötschel and Wakabayashi (1989) to solve cluster editing; the triangular constraints ensure that all within-cluster distances are used to compute the objective function, ignoring between-cluster distances. In other words, they ensure that for elements x_i, x_j that belong to the same anticluster, we obtain $x_{ij} = 1$; for elements x_i, x_j that do not belong to the same anticluster, we obtain $x_{ij} = 0$. The triangular constraints thereby enforce the anticlustering restrictions (1) and (2), i.e., they ensure that each element is assigned to exactly one anticluster. Constraint (18) enforces anticlustering restriction (3) that each anticluster consists of the same number of elements $\frac{n}{K}$. This is accomplished by uniting each element with $\frac{n}{K} - 1$ other elements in the same anticluster. Constraint (19) ensures that the decision variables x_{ij} are binary.

In the `anticlust` package, one of the commercial solvers `gurobi` or `CPLEX`, or the open source GNU linear programming kit can be used to solve the ILP formulation (14) - (19).

3.3 Additional restrictions on the ILP

To expand its applicability to larger problem sizes, we added additional constraints to the ILP formulation of distance anticlustering. The constraints result from two considerations that in conjunction lead to a redefinition of the cost function w:

1. The run time of the weighted cluster editing ILP is improved if there is an uneven distribution of edge weights (Böcker & Baumbach, 2013; Böcker et al., 2011)

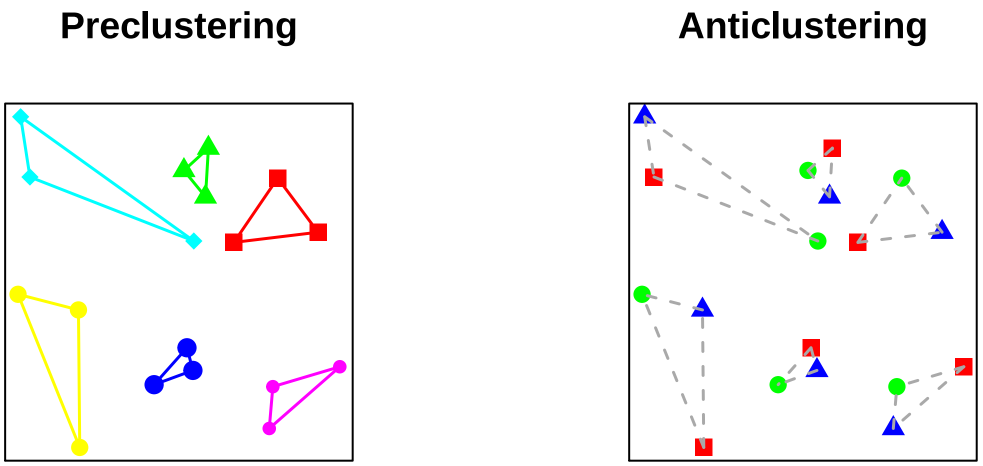

Figure 2: The left-hand plot shows six preclusters with a respective size of three elements. The right-hand plot shows the optimal anticlustering under the preclustering restrictions. Elements that are part of the same precluster cannot be part of the same anticluster

2. It is possible to prevent very similar elements from joining the same anticluster without impairing the quality of the solution strongly

In the cluster editing ILP framework, setting $w(ij) = -\infty$ prevents two elements x_i and x_j from joining the same anticluster (Böcker et al., 2011). If edge weights for similar elements are redefined like this, we induce a strong enevenness in edge weights.

Therefore, the `anticlust` package realizes a preprocessing step in which very similar elements are grouped into *preclusters*; elements of a precluster are prevented from joining the same anticluster thereafter. The preclustering step is formalized by the ILP defined in (20) - (25).

$$\text{Minimize} \sum_{1 \leq i < j \leq n} w(ij)\, x_{ij} \tag{20}$$

$$-x_{ij} + x_{ik} + x_{jk} \leq 1, \qquad \forall\, 1 \leq i < j < k \leq n, \tag{21}$$

$$x_{ij} - x_{ik} + x_{jk} \leq 1, \qquad \forall\, 1 \leq i < j < k \leq n, \tag{22}$$

$$x_{ij} + x_{ik} - x_{jk} \leq 1, \qquad \forall\, 1 \leq i < j < k \leq n, \tag{23}$$

$$\sum_{1 \leq i < j \leq n} x_{ij} + \sum_{1 \leq k < i \leq n} x_{ki} = K - 1, \qquad \forall i \in \{1, ..., n\} \tag{24}$$

$$x_{ij} \in \{0, 1\},\ \forall\, 1 \leq i < j \leq n \tag{25}$$

By minimizing the distance objective, the preclustering step solves weighted cluster editing under the restriction of a cluster size K. That is, each precluster contains as many elements as there are anticlusters. For $K = 2$, the preclustering corresponds to the minimum weight perfect matching problem (Gerards, 1995). The left-hand plot in Figure 2 illustrates the preprocessing step for $K = 3$ and $n = 18$.

In a second step, the cost function is redefined as follows:

$$w(ij) = \begin{cases} -\infty & \text{if } x_{ij} = 1 \\ d_{ij} & \text{if } x_{ij} = 0 \end{cases}$$

Using the redefined edge weights, we solve the distance anticlustering in (14) - (19). We prevent elements from the same precluster from joining the same anticluster; the results of an anticlustering employing the preclustering restrictions is shown in the right-hand plot of Figure 2. Note that the optimal solution sometimes requires to join elements that are part of the same precluster within the same precluster. Therefore, the preprocessing sometimes precludes an optimal solution.

3.4 The heuristic approach

- Simulated annealing
- Neighborhood of elements, preclustering

4 References

Arabie, P., & Hubert, L. J. (1996). An overview of combinatorial data analysis. In P. Arabie, L. J. Hubert, & G. De Soet (Eds.), *Clustering and classification* (pp. 5–63).

Bacon, D. R. (2001). An evaluation of cluster analytic approaches to initial model specification. *Structural Equation Modeling*, *8*(3), 397–429.

Bansal, N., Blum, A., & Chawla, S. (2004). Correlation clustering. *Machine Learning*, *56*(1-3), 89–113.

Böcker, S., & Baumbach, J. (2013). Cluster editing. In *Conference on Computability in Europe* (pp. 33–44). Springer.

Böcker, S., Briesemeister, S., & Klau, G. W. (2011). Exact algorithms for cluster editing: Evaluation and experiments. *Algorithmica*, *60*(2), 316–334.

Gerards, A. M. H. (1995). Matching. In M. O. Ball, T. L. Magnanti, C. L. Monma, & G. L. Nemhauser (Eds.), *Network models* (pp. 135–224).

Grötschel, M., & Wakabayashi, Y. (1989). A cutting plane algorithm for a clustering problem. *Mathematical Programming*, *45*(1-3), 59–96.

Guha, S., Rastogi, R., & Shim, K. (1998). CURE: An efficient clustering algorithm for large databases. In *ACM sigmod record* (Vol. 27, pp. 73–84). ACM.

Jain, A. K. (2010). Data clustering: 50 years beyond k-means. *Pattern Recognition Letters*, *31*(8), 651–666.

Lorena, L. H. N., Quiles, M. G., Carvalho, A. C. P. de L. F. de, & Lorena, L. A. N. (2018). Preprocessing technique for cluster editing via integer linear programming. In D.-S. Huang, V. Bevilacqua, P. Premaratne, & P. Gupta (Eds.), *Intelligent computing theories and application* (pp. 287–297). Cham: Springer International Publishing.

Mertens, S. (2001). A physicist's approach to number partitioning. *Theoretical Computer Science*, *265*(1-2), 79–108.

Miyauchi, A., & Sukegawa, N. (2015). Redundant constraints in the standard formulation for the clique partitioning problem. *Optimization Letters*, *9*(1), 199–207.

Rahmann, S., Wittkop, T., Baumbach, J., Martin, M., Truss, A., & Böcker, S. (2007). Exact and heuristic algorithms for weighted cluster editing. In *Computational systems bioinformatics: (Volume 6)* (pp. 391–401). World Scientific.

Schaeffer, S. E. (2007). Graph clustering. *Computer Science Review*, *1*(1), 27–64.

Späth, H. (1986). Anticlustering: Maximizing the variance criterion. *Control and Cybernetics*, *15*(2), 213–218.

Valev, V. (1998). Set partition principles revisited. In *Joint IAPR international workshops on statistical techniques in pattern recognition (SPR) and structural and syntactic pattern recognition (SSPR)* (pp. 875–881). Springer.

Wittkop, T., Emig, D., Lange, S., Rahmann, S., Albrecht, M., Morris, J. H., … Baumbach, J. (2010). Partitioning biological data with transitivity clustering. *Nature Methods*, *7*(6), 419–420.