Supplementary materials to TITLE

Martin Papenberg

Standard algorithm for anticlustering

To assign samples to batches, we use anticlustering. Anticlustering is an optimization method that is characterized by (a) an objective function that quantifies the balance among batches, and (b) an algorithm that conducts the batch assignment in such a way that balance among batches is maximized via the objective function. Anticlustering owes its name to the fact that the objective functions it uses are the reversal of criteria used in cluster analysis. For example, Späth (1986) already recognized that by maximizing instead of minimizing the k-means criterion (the "variance"), he was able to create groups that are similar to each other, and presented it as an improvement over the more intuitive random assignment.

In our application, we optimize the (a) diversity objective using (b) the exchange algorithm by Papenberg and Klau (2021). This constellation of objective and algorithm is also the default method implemented in the R package `anticlust` (Papenberg 2024). The diversity is defined as the total sum the within-cluster sums of dissimilarities (Brusco, Cradit, and Steinley 2020). When assigning samples in such a way that the diversity is maximized, we simultaneously maximize similarity between batches. Papenberg and Klau (2021) referred to the maximization of the diversity as "anticluster editing" because the minimization of the diversity is also well-known from the area of cluster analysis—under the term "cluster editing" (Shamir, Sharan, and Tsur 2004; Böcker, Briesemeister, and Klau 2011).

The anticlustering exchange algorithm takes as input (1) a distance matrix, quantifying the pairwise dissimilarity among samples, and (2) the number of batches K. As measure of dissimilarity, we use the common Euclidean distance, which transfers the features of our samples to pairwise distances. It is defined as

$$d(x, y) = \sqrt{\sum_{i=1}^{M} (x_i - y_i)^2}$$

where M is the number of numeric features describing two samples $x = (x_1, \ldots, x_M)$ and $y = (y_1, \ldots, y_M)$. When samples are described by two features, the Euclidean distance corresponds to the geometric, "straightline" distance between two points in a two-dimensional space; more similar data points are closer to each other. Figure 1 illustrates the computation of the Euclidean distance and the diversity for the numeric features BMI and age that were used in our motivating application. For categorical variables, we use binary coding before including them in the computation of the Euclidean distance. Table 1 illustrates binary coding for the feature race in our data set, which had five unique values: (a) Asian, (b) White, (c) Native Hawaiian or Other Pacific Islander, (d) Black, (e) Native American. Such a categorical variable can be recoded into four binary variables to maintain the information included in the original variable.

Table 1: Illustrate the recoding of the variable Race using four binary variables.

Race	Black	Native American	Pacific Islander	White
Black	1	0	0	0
Pacific Islander	0	0	1	0
Native American	0	1	0	0

Race	Black	Native American	Pacific Islander	White
White	0	0	0	1
Asian	0	0	0	0

The exchange algorithm that maximizes the diversity consists of two steps: an initialization step and an optimization step. As initialization, it randomly assigns samples to K equal-sized batches. In principle, unequal-sized batches would also be possible, but equal-sized batches were required in the current application. After initialization, the algorithm selects the first sample and checks how the diversity would change if the sample were swapped with each sample that is currently assigned to a different batch. After simulating each exchange—that is $N - \frac{N}{K}$ exchanges—it realizes the one exchange that increases the diversity the most. It does not conduct an exchange if no improvement in diversity is possible. This procedure is repeated for each sample, leading to the evaluation of $N \cdot \frac{N}{K}$ exchanges in total; it terminates after the last sample was processed. The procedure might also restart at the first element and reiterate through all samples until no exchange leads to an improvement any more, i.e., until a local maximum is found. In `anticlust`, we also implemented this local maximum search, which corresponds to the algorithm LCW by Weitz and Lakshminarayanan (1998). For better results, it is also possible to restart the search algorithm multiple times using different (random) initializations (Späth 1986).

Including must-link constraints with anticlustering

In our application, samples belonging to the same patient were required to be assigned to the same batch; we refer to a set of samples that must be assigned to same batch as a *must-link group*. For the data set that resembled our actual application, we simulated 370 samples from 191 unique patients. Most patients (n = 100) were included with 1 sample. The remaining 270 patients were part of a must-link group, with group sizes varying between 2 and 8: 47 groups of size 2, 13 groups of size 3, 24 groups of size 4, 3 groups of size 5, 3 groups of size 6, and 1 group of size 8.

To include must-link constraints with anticlustering, we use a downscaled data set where each patient constitutes a single unit in the anticlustering process. Here, the effective sample size for anticlustering with must-link constraints is 191 instead of 370. However, some adjustments of the exchange method are required to ensure (a) we still obtain a valid partitioning regarding the size constraints (equal-sized batches) and (b) the diversity is computed correctly during optimization. We therefore had to adjust both the initialization phase as well as the optimization phase of the exchange algorithm.

During initialization, we first assign all samples to a batch that are part of a must-link group (270 samples). Each must-link group must be assigned in its entirety to one of the K batches while the maximum capacity of each batch must not be exceeded. To solve this problem, we assign a weight to each must-link group, corresponding to the number of samples it contains. Hence, there are K batches each with capacity $\frac{N}{K}$, which have to be filled with samples. The sum of the weights of the must-link groups in each batch must not exceed its capacity. Using this conceptualization, the initialization step corresponds to a bin packing problem, which is one the classical NP complete problem in computer science (Garey and Johnson 1979). Many optimal and heuristic algorithms have been developed to address it. We use a randomized fit heuristic to fill the batches: For each must-link cluster, we iterate through the K batches in random order and assign it to the first batch where it fits. The process is expected to evenly distribute the must-link clusters among batches, and the random component is particularly useful if we use multiple restarts for the exchange algorithm. After assigning the must-link clusters to batches, the remaining samples can be assigned randomly to fill the remaining space. Note that our randomized fit algorithm is a heuristic that may not find an assignment of must-link groups to batches, even if one is theoretically available. If the heuristic indicates that the batches cannot hold the must-link groups, we therefore use an optimal algorithm based on integer linear programming as fallback option, which allows us to verify if the constraints really cannot be fulfilled (Martello and Toth 1990).

The optimization phase of the exchange algorithm also uses the downscaled data set where each must-link group corresponds to a single unit of analysis. For example, in our application, the original data set consisted

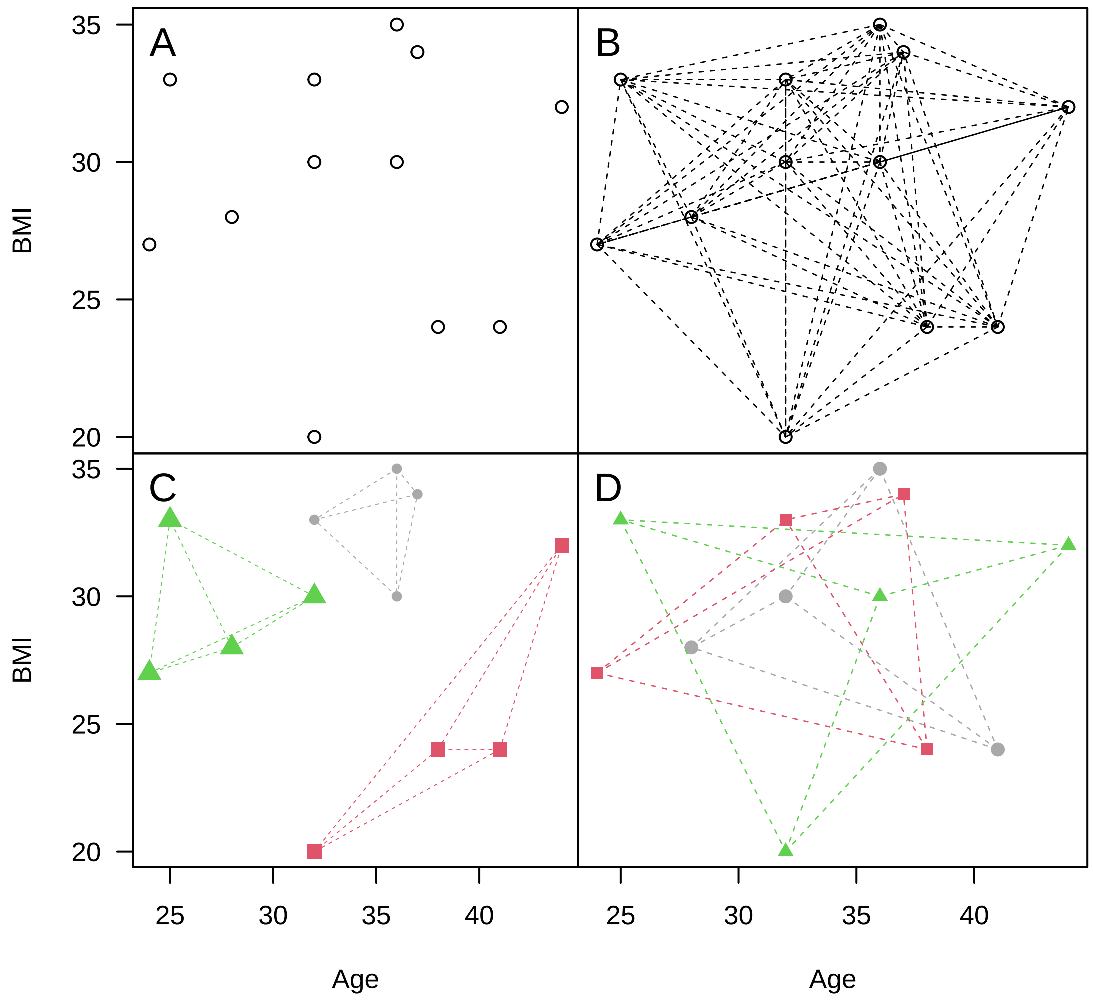

Figure 1: Illustrates the conversion from numeric features to Euclidean distance, and (anti)clustering assignments based on minimum and maximum diversity using the Euclidean distance. Panel A illustrates the BMI and age of twelve women in our synthetic data in a scatter plot. Panel B represents the Euclidean distances between features as a straight line in the two-dimensional space. The Euclidean distance is proportional to the length of the connecting lines in panel B. Panel C illustrates a clustering assignment of the 12 data points to $K = 3$ equal-sized groups via *minimum* diversity. Panel D illustrates an anticlustering assignment of the 12 data points to $K = 3$ equal-sized groups via *maximum* diversity. The diversity is computed as the sum of within-(anti)cluster distances, which are highlighted in Panel C and Panel D through connecting lines.

of 370 samples, but the optimization algorithm works on a data set of 191 unique patients. To obtain the reduced distance matrix that preserves all information of the original distance matrix, we can sum all pairwise distances between elements in different must-link clusters (Böcker, Briesemeister, and Klau 2011). This transformation preserves the computation of the diversity as compared to using the full data set. The exchange heuristic is then conducted on the merged elements. During the exchange process, we only exchange must-link clusters of the same size to ensure that the cardinality constraints are still respected in the end (i.e., equal-sized groups).

Simulation Study

We conducted a simulation study to illustrate the usefulness of anticlustering for batch assignment. The simulation followed two goals. First, it compared anticlustering with the OSAT package, which is currently the gold standard for automated batch assignment (citation). Second, it investigated whether the quality of batch assignment is reduced with must-link constraints as compared to an unconstrained assignment. We generated 10000 data sets, which were all processed via (a) the default OSAT algorithm (optimal shuffle with 5000 repetitions), (b) unconstrained anticlustering and (c) anticlustering subject to must-link constraints. In each data set, we randomly determined the number of categorical variables (2-5), the number of classes per variable (2-5), the total sample size (between 50 and 500) and the number of batches (2, 5, or 10 equal-sized batches). Must-link constraints were randomly generated in such a way that the distribution of constraints resembled our motivating application: 58% of all elements had no must-link partner; 29% had 1 must-link partner; 10% had 2 must-link partners, and 3% had 3 or more must-link partners. The code and data to reproduce the simulation and analysis is openly available via XXX (**TODO**).

For each simulation run, we computed χ^2-tests to assess the imbalance among batches for each of the 2-5 variables, for each of the three competing methods. We stored the p value associated with each test. A higher p value indicates that there is less imbalance among batches, i.e., that the batches are more similar. The simulation revealed that in 84% of all variable comparisons, balance among batches was better when using anticlustering as compared to OSAT. Balance was equal in 14% of all comparisons, and only in 1% OSAT outperformed anticlust. These results clearly underline the power of anticlustering for batch assignment. Moreover, the anticlustering assignment that was subjected to must-link constraints on average achieved 99.8% of the objective value of the unconstrained assignment. Hence, must-link constraints are not only desirable from a user's point of view, but they also do not decrease batch balance considerably; in 54% of all cases, balance was not at all reduced by the constraints. Figure 2 illustrates the average balance among batches in dependence of the factors that varied between data sets. Must-link constraints primarily reduced balance in smaller data sets, but did not strongly affect the balance in larger data sets. Remarkably, the constrained anticlustering assignment led to better balance than the OSAT assignment that did not employ any constraints. Figure 2 also shows increasing the number of variables as well as increasing the number of categories per variables posed severe challenges for OSAT, but hardly affected anticlustering.

Optimal algorithm for anticlustering under cannot-link constraints

- Solving anticlustering optimally (i.e., finding a batch assignment with globally optimal value in diversity) under must-link constraints is possible using the ILP by Papenberg and Klau (2021)
- We have to adjust the input matrix: Samples that must be linked receive sufficiently large pairwise distance; in this case, a globally optimal solution must pair these samples in the same batch (if the constraints can be fulfilled at all)
- We investigate the feasibiliy of the optimal approach, which solves a computationally difficult (NP hard) problem (it seems to scale to about 40-50 samples; for larger data sets, a heuristic is required)
- We can use the optimal algorithm to validate the quality of our heuristic (how often does it find the globally optimal solution)

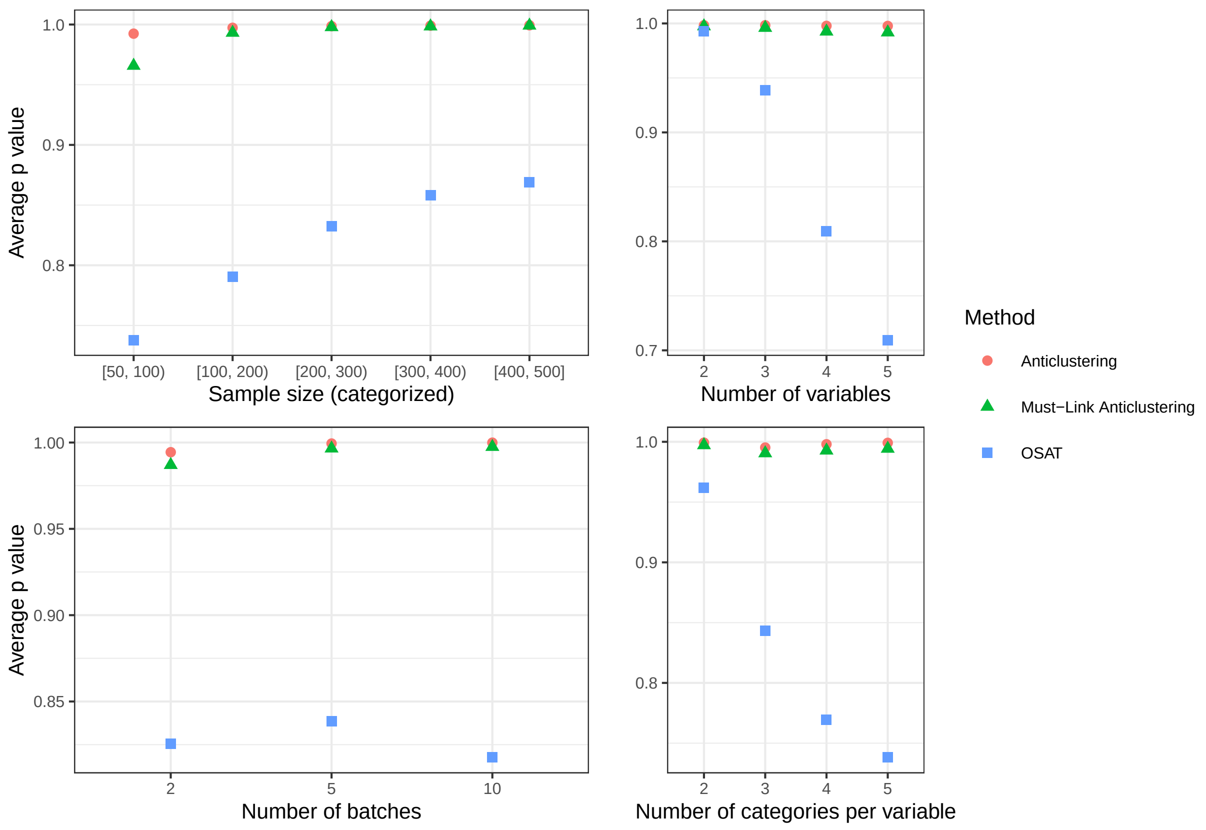

Figure 2: Average p values in dependence of the factors that varied in the simulation study. Higher p values indicate better balance. Anticlustering maintained a comparable level of balance in all conditions. OSAT's performance decreased with increasing number of variables and number of categories per variable.

References

Böcker, Sebastian, Sebastian Briesemeister, and Gunnar W Klau. 2011. "Exact Algorithms for Cluster Editing: Evaluation and Experiments." *Algorithmica* 60: 316–34. https://doi.org/10.1007/s00453-009-9339-7.

Brusco, Michael J, J Dennis Cradit, and Douglas Steinley. 2020. "Combining Diversity and Dispersion Criteria for Anticlustering: A Bicriterion Approach." *British Journal of Mathematical and Statistical Psychology* 73 (3): 375–96. https://doi.org/10.1111/bmsp.12186.

Garey, Michael R, and David S Johnson. 1979. *Computers and Intractability: A Guide to the Theory of NP-completeness*. New York: Freeman.

Martello, S, and P Toth. 1990. *Knapsack Problems: Algorithms and Computer Implementations*. Wiley.

Papenberg, Martin. 2024. "Anticlust: Subset Partitioning via Anticlustering." https://cran.r-project.org/pa ckage=anticlust.

Papenberg, Martin, and Gunnar W Klau. 2021. "Using Anticlustering to Partition Data Sets into Equivalent Parts." *Psychological Methods* 26 (2): 161–74. https://doi.org/10.1037/met0000301.

Shamir, Ron, Roded Sharan, and D Tsur. 2004. "Cluster Graph Modification Problems." *Discrete Applied Mathematics* 144: 173–82. https://doi.org/10.1016/j.dam.2004.01.007.

Späth, H. 1986. "Anticlustering: Maximizing the Variance Criterion." *Control and Cybernetics* 15 (2): 213–18.

Weitz, RR, and S Lakshminarayanan. 1998. "An Empirical Comparison of Heuristic Methods for Creating Maximally Diverse Groups." *Journal of the Operational Research Society* 49 (6): 635–46. https://doi.org/10.1057/palgrave.jors.2600510.