Supplementary materials to TITLE

Martin Papenberg

Example Application

The ENACT project is concerned with researching endometriosis disease. One of the ENACT projects required to assign 320 cell samples of women either carrying the disease or not to 20 batches. To illustrate the application, we prepared a synthetic data set that resembles the actual data set, which is not disclosed for reasons of medical confidentiality. Women in varying disease stages provided a varying number of samples. All women who did not carry the disease ($n = 50$) provided exactly one sample, while all women who carried the disease ($n = 89$) provided multiple samples: 40 patients provided 2 samples, 25 patients provided 3 samples, 11 patients provided 4 samples, 10 patients provided 5 samples, and 6, 7 or 8 samples were provided by one patient, respectively. In total, the data set consists of 320 samples from 139 unique individuals.

For the assignment, we strived for balance with regard to four categorical variables: is disease present (yes or no); stage of disease ("none", "Stage I or II", "Stage III or IV"); clinical site (University of California SF, or other); phase (PE or SE) (what is this?). Other demographic variables such as age and BMI were also assessed but not deemed critical as covariates for the batch assignment. Post-hoc checks however revealed that no significant discrepancies occurred in these other variables, despite not being included.

For the purpose of illustration, we first applied an unconstrained anticlustering assignment and OSAT (see Table 1 and 2). Both methods successfully balanced all four variables among batches (all ps > .99). While the assignment was conducted using the 320 samples—as opposed to the 139 individual patients—as input, we also verified satisfactory balance on the level of individuals. Table 3 illustrates the results of constrained anticlustering, which through the 2PML algorithm ensured that samples belonging to the same patient were assigned to the same batch. Note that the must-link constraints put rather severe restrictions to the assignments that were possible; for example, up to 50% of a batch had to be occupied with the samples of a single patient. Still, using 2PML, batches turned out to be highly balanced (ps > .99 on batch level and ps > .98 on individual level). All assignments and results are reproducible via the accompanying online repository (link).

Methods

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. 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 (Steinley, 2006). Brusco et al. (2020) recognized that other objective functions known from cluster analysis can also be implemented in the context of anticlustering.

In our application, we optimized the diversity objective to maximize similarity among batches. While the diversity is technically a measure of within-batch heterogeneity, its maximization simultaneously leads to minimal difference between the distribution of the input variables among batches (Feo & Khellaf 1990; cf. Papenberg, 2024). 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 et al., 2004; Böcker et al., 2011). The diversity is computed on the basis of a measure of pairwise dissimilarity among samples. In particular, it is defined as the overall sum of all dissimilarities among samples that are assigned to the same batch (Brusco et al., 2020). Hence, the diversity is not directly computed on the basis of the samples' features, but instead it relies on distance measure that is computed on the basis of the features, for each pair of samples. In the context of anticlustering, the Euclidean distance is the most common measure that translates features to pairwise dissimilarities (Gallego et al. 2013; Papenberg & Klau, 2021). However, using other distance measures such as the squared Euclidean distance is also possible (Brusco et al., 2020). The Euclidean distance is defined as

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

When samples are described by two features, the Euclidean distance corresponds to the geometric, "straight line" distance between two points in a two-dimensional space; more similar data points are closer to each other (see Figure 1). For categorical variables, we use binary coding before including them in the computation of the Euclidean distance (see Table 2). An anticlustering algorithm assigns samples to batches in such a way that the objective function—here, the diversity—is maximized. Anticlustering usually employs heuristic optimization algorithms (Yang et al., 2022). While heuristics generally provide satisfying results in the context of anticlustering (Papenberg & Klau, 2021), they do not guarantee to find the globally best assignment among all possibilities. In principle, enumerating all possible assignments is a valid strategy to obtain an optimal assignment. However, this approach quickly becomes impossible due to an exponential growth of the way in which assignments can be conducted (Papenberg & Klau, 2021). Moreover, because anticlustering problems (with the exception of some special cases) are also NP-hard (Feo & Khellaf, 1990), there is probably no algorithm that identifies the globally best assignment without considering all possibilities (at least in the worst case). In practice, heuristics are therefore indispensable. In the following, we present an anticlustering heuristic that is generally applicable to large data sets, and outline how we included must-link constraints with this algorithm.

We used an heuristic exchange algorithm to maximize the diversity (Späth 1986; Papenberg & Klau, 2021; Weitz & Lakshminarayanan, 1998). It consists of two steps: an initialization step and an optimization step. As initialization, it randomly assigns samples to equal-sized batches. In principle, unequal-sized batches would also be possible, but equal-sized batches were required in the current application (and in general, this requirement is most common. 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, 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 and 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).7 For better results, it is also possible to restart the search algorithm multiple times using different (random) initializations (Späth 1986). For the anticlustering assignment illustrated in Table 1, we employed 10 repetitions of the local maximum search LCW, using the squared Euclidean distance as measure of pairwise dissimilarity. To ensure comparable weight of the three features, we also applied a standardization of the input variables before the distance was computed. Using a standardization is recommended if the variables differ strongly in their ranges (Papenberg, 2024).

Table 1: Illustrates the recoding of the categorical variable Stage using three binary variables.

Stage	None	StageI_II	StageIII_IV
None	1	0	0
StageI_II	0	1	0
StageIII_IV	0	0	1

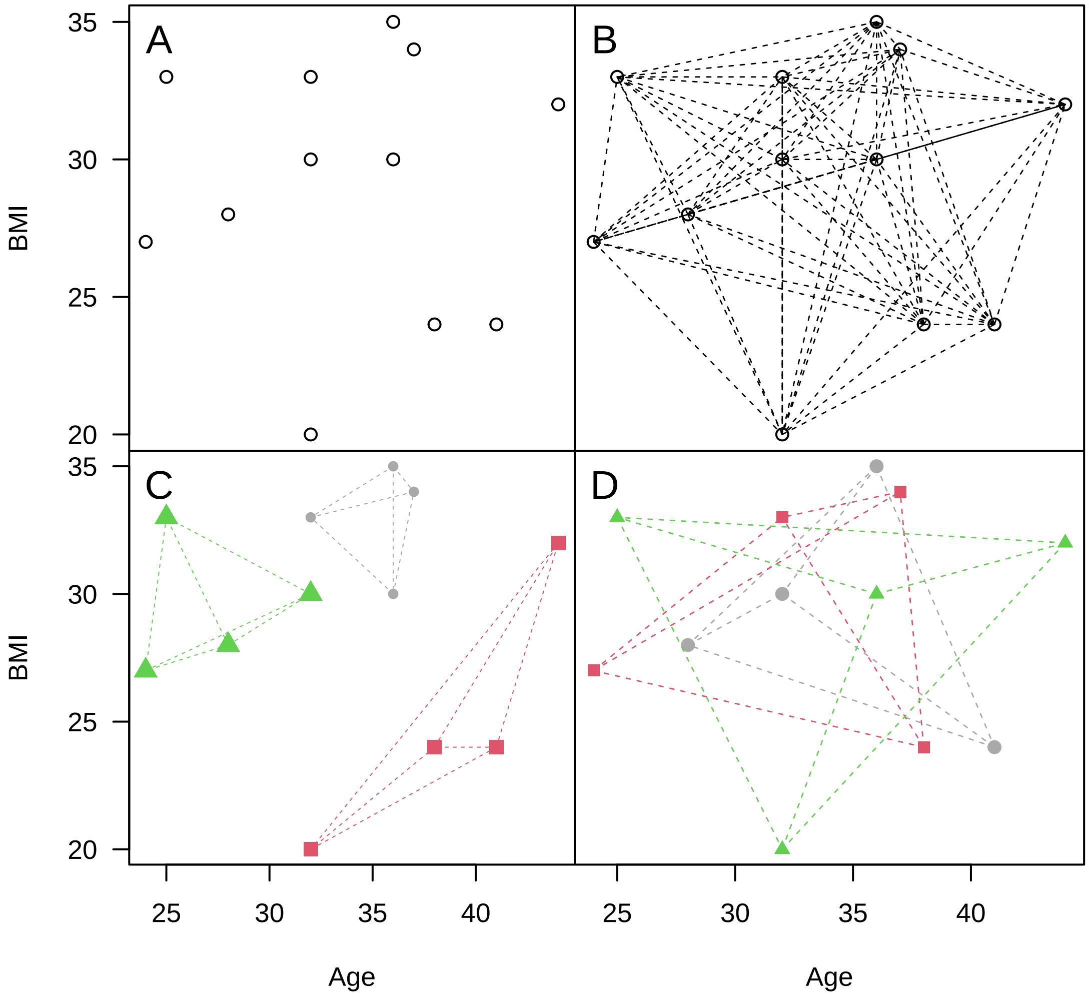

Figure 1: Illustrates the conversion from numeric features to Euclidean distance, and (anti)clustering assignments based on minimum and maximum diversity using the Euclidean distance. Panel A illustrates the BMI and age of twelve women in our synthetic data in a scatter plot. Panel B represents the Euclidean 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. Maximizing the diversity simultaneously leads to similar distribution of the input features among batches.

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 the same batch as a must-link clique. In our application, not all samples were part of a clique. For the synthetic data set that resembled our actual application, we simulated 370 samples from 191 unique patients. To include the must-link constraints with anticlustering, we basically use a downscaled data set where each unique patient—but not each single sample—constitutes a unit in the anticlustering process. Hence, in our application, the effective sample size was 191 instead of 370. 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 clique (270 samples). Each clique must be assigned completely to one of the batches and samples within a clique must not be split apart. At the same time, the maximum capacity of each batch must not be exceeded. Using this conceptualization, the initialization step corresponds to a bin packing problem, which is one the classical NP complete problem in computer science (Garey & Johnson, 1979). That is, we assign a weight to each clique, corresponding to the number of samples it contains. When filling batches, the sum of the weights of the cliques in each batch must not exceed its capacity. Many optimal and heuristic algorithms have been developed to address such a bin packing problem. As the default method, we use a randomized first fit heuristic to fill batches: For each must-link clique, we iterate through all batches in random order and assign it to the first batch where it fits. The process is expected to evenly distribute the must-link cliques among batches. This random component is particularly useful if we use multiple restarts of the optimization algorithm. After assigning the must-link cliques to batches, the remaining samples can be assigned randomly to fill the remaining space. Note that our randomized first 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 a fallback option, which allows us to verify if the constraints really cannot be fulfilled. To this end, we implemented an adaptation of the standard bin packing ILP model by Martello and Toth, (1990, 221). It is given as

$$
\begin{aligned}
&\text{minimize} && \sum_{1 \leq i \leq K} \sum_{1 \leq j \leq n} x_{ij} && && (1)\\
&\text{subject to} && \sum_{j=1}^{K} w_j x_{ij} \leq c_i && i = (1, \ldots, n) && (2)\\
& && \sum_{i=1}^{n} x_{ij} = 1 && j = (1, \ldots, K) && (3)\\
& && x_{ij} \in \{0, 1\} && i = (1, \ldots, K), j = (1, \ldots, n) && (4)
\end{aligned}
$$

The number of must-link cliques is given by n. The model has decision variables x_{ij} to encode whether clique j ($j = 1, \ldots, n$) is assigned to batch i ($i = 1, \ldots, K$). It uses K values c_i to represent the capacity of each batch; in the default case of equal-sized batches, we have $c_i = \frac{N}{K}$ for each batch. It uses n values w_j to encode the weight of each clique, i.e., the number of samples it represents that must be assigned to the same batch in order to fulfil the must-link constraints. Constraint (2) realizes that the weight of each batch is not exceeded; constraint (3) realizes that each clique is assigned to exactly one batch. Note that during the initialization step that assigns cliques to batches, we only need to test if the constraints (2) and (3) can be fulfilled at all; any feasible assignment is equally valid. For this reason, the objective function (1) is chosen to be constant for each assignment that satisfies the constraints (n). It does not actually contribute to solving the problem, and the model only test if the must-link constraints can be fulfilled.

Like the initialization step, the optimization step of the exchange algorithm uses a downscaled data set where each patient rather than each sample corresponds to a unit of analysis. However, to obtain valid results according to the original data set (which incorporates all samples), an adjustment to the data input is needed. In particular, we must change the matrix of pairwise dissimilarities: To obtain a reduced distance matrix that preserves all information of the original distance matrix, we sum up all pairwise distances between samples in different cliques. In the context of maximizing the diversity, this transformation sufficiently preserves the relevant information in the original distance matrix (Böcker et al., 2011). Using the initial assignment and the reduced distance matrix, we apply the same exchange algorithm as we described for the unrestricted anticlustering application, with one adjustment: During the exchange process, we only exchange cliques of the same size (e.g., patients providing the same number of samples) to ensure that the cardinality constraints are respected throughout (i.e., usually equal-sized batches).

Optimal anticlustering using must-link constraints

As mentioned above due the computational complexity of anticlustering, batch assignment problems are usually tackled using heuristic algorithms. Still, for some problem constellations, in particular when N is not large, it is possible to employ optimal algorithms that find the globally best batch assignment. Papenberg and Klau (2021) presented an ILP model to find globally optimal batch assignments for the diversity objective. It can be used to solve problem instances of up to about $N = 30$ in an acceptable running time; Schulz (2022;). In this paper, we extend the model by Papenberg and Klau (2021) by allowing it to include must-link constraints. The extension is actually quite straight forward: To induce must-link constraints in the context of an optimal algorithm, it is sufficient to adjust the distance matrix used as input. Whenever the pairwise distance between two samples it set to ∞—and the set of must-link constraints can be fulfilled—any globally optimal assignment will place these samples in the same batch, because the objective value associated with such an assignment is necessarily better than that of an assignment that places them in different batches. This feature of adjusting the data input has long been used in the context of optimal algorithms for cluster editing (Böcker et al. 2011). Including must-link constraints in the anticlustering problem specification increases the number of samples that can be processed. In our online supplementary materials, we show that up to about 40-50 samples can be processed in a time limit of 1800 seconds on a personal computer; after that, the running time is expected to increase exponentially with increasing number of samples.

Simulation Study

For the simulation, we generated 2000 data sets. Data sets were processed via (a) OSAT, (b) PSBA, (c) unconstrained anticlustering and (d) anticlustering subject to must-link constraints. Because the OSAT method is only applicable to categorical variables, we used categorical variables in our simulation. For anticlustering and for PSBA, the categorical variables were binary coded before the methods were applied (see Table 2). For each data set, we randomly determined the number of categorical variables (2-5), the number of classes per variable (2-5; the distribution of classes was uniform), the total sample size N (between 50 and 500) and the number of batches K (2, 4, or 10 equal-sized batches). PSBA was only applied for $K = 2$ and $K = 4$ ($n = 1375$ data sets) because the authors' implementation only allows the assignment to a maximum of four batches. Must-link constraints were generating a random integer between 1 and N (with equal probability), and using the resulting numbers as must-link grouping variables. This rule resulted in a distribution of constraints that 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.

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 83% of all variable comparisons, balance among batches was better when using anticlustering as compared to OSAT. Balance was equal in 16% of all comparisons, and only in 1% OSAT outperformed `anticlust`. In 51% of all variable comparisons, balance was better when using anticlustering as compared to PSBA. Balance was equal in 16% of all comparisons, and in 34% PSBA outperformed anticlustering. Figure 2 illustrates the average p-values in dependence of the number of variables (M) and the number of batches (K). The online supplement also includes additional Figures illustrating average p-values in dependence of the other factors that varied in the simulation (the sample size N and the number of classes per categorical variables; **TODO**). Figure 2 shows that increasing the number of variables posed severe challenges for OSAT, but hardly affected anticlustering. PSBA also showed decreased performance with an increasing number of variables, but less so than OSAT.

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 57% of all cases, balance was not at all reduced by the constraints. Remarkably, the constrained anticlustering assignment led to better balance than the OSAT and PSBA assignments that did not employ any constraints (see Figure 2).

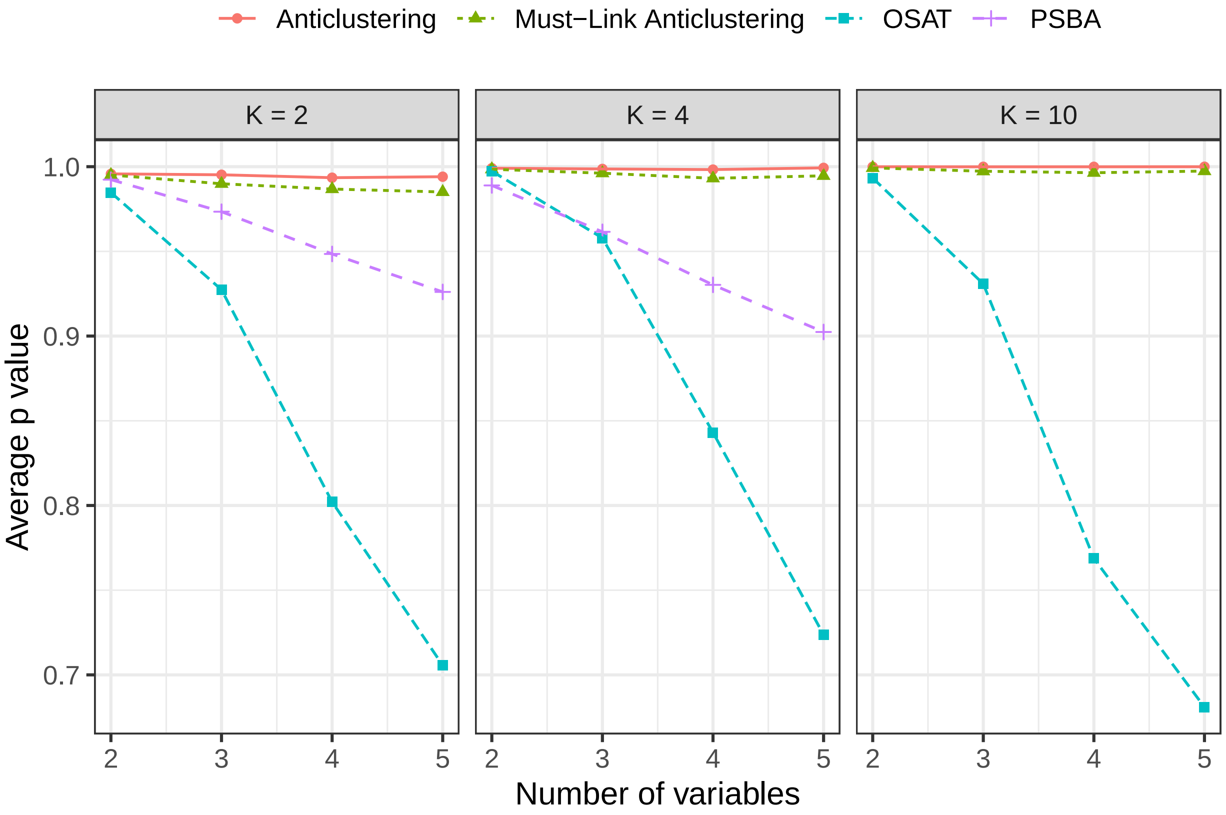

Figure 2: Average p values in dependence of the number of batches and the number of variables. Higher p values indicate better balance. Anticlustering maintained a comparable level of balance in all conditions. OSAT's performance decreased with increasing number of variables most strongly.

References

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