Extending Standard Cluster Algorithms to Allow for Group Constraints*

Friedrich Leisch[1] and Bettina Grün[2]

[1] Institut für Statistik, Universität München, Ludwigstraße 33, D-80539 München, Germany; Friedrich.Leisch@stat.uni-muenchen.de
[2] Institut für Statistik, Technische Universität Wien, Wiedner Hauptstraße 8-10/1071, A-1040 Wien, Austria; Bettina.Gruen@ci.tuwien.ac.at

Summary. This paper demonstrates how standard cluster algorithms like K-means or partitioning around medoids can be modified such that the final solution fulfills group constraints, which specify that certain data points must be or may not be in the same cluster. An extensible software implementation for the R statistical computing environment is presented that allows user-specified group constraints for clustering with respect to arbitrary distance measures. Finally we discuss applications of the methodology to market segmentation of household shopping basket panel data and model diagnostics for finite mixture models.

Key words: cluster analysis, grouped data, R, flexclust

1 Introduction

While statistical models for stratified data with nested groupings like mixed effects models with complicated random effect structures have been available for some time now, clustering data with grouping information has received only little attention in the literature. This is insofar surprising, as the main task of clustering is to group data, and incorporating prior grouping information into the clustering procedure seems rather natural. An exception is model-based clustering, especially because several mixture model classes are not identifiable without repeated measurements and hence grouping information is a necessity there [MP00, Grü02]. As a consequence mixture modelling software like R [R D05] package `flexmix` [Lei04] can deal with grouping information that specifies which groups of observations must be in the same cluster.

Model-free clustering procedures like K-centroids cluster analysis [KCCA, e.g., Lei06] are almost never combined with group constraints, and especially in the statistical literature we are not aware of any publications. An exception in the machine

[*] This paper was published as: Friedrich Leisch and Bettina Grün. Extending standard cluster algorithms to allow for group constraints. In Alfredo Rizzi and Maurizio Vichi, editors, Compstat 2006—Proceedings in Computational Statistics, pages 885–892. Physica Verlag, Heidelberg, Germany, 2006.

learning community is the work by [WCRS01], who modify K-means by a greedy search step for the optimal cluster assignment under two types of group constraints. We extend this approach to another class of cluster algorithms, namely exchange methods, and we show how the greedy search can be replaced by a locally optimal step in each iteration by using the global optimizer of a linear sum assignment problem.

2 Clustering with Group Constraints

Assume we are given a data set $X_N = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ of size N, and let $d(\mathbf{x}, \mathbf{y})$ denote a distance measure between points $\mathbf{x}$ and $\mathbf{y}$. The objective of centroids-based partitioning cluster analysis is to find a set of K centroids C_K such that the average distance of each point to the closest centroid is minimal,

$$D(X_N, C_K) = \frac{1}{N} \sum_{n=1}^{N} d(\mathbf{x}_n, c(\mathbf{x}_n)) \to \min_{C_K},$$

where $c(\mathbf{x})$ is the centroid $\mathbf{c} \in C_K$ closest to $\mathbf{x}$. Euclidean distance is certainly the most popular choice for d, but the methods and software tools presented in this paper work for distance measures in general like other Minkowski metrics, transformations of similarity measures like correlation, etc.; see [Lei06] for examples and implementation details.

Even for very simple distance measures no closed form solution for the K-centroids cluster problem exists and iterative estimation procedures have to be used. Two classes of KCCA algorithms are most popular within the statistical community[3]: variations of the K-means algorithm in its general form [e.g., Mac67], which use the whole data set in each iteration of the algorithm; and exchange algorithms like the one described in [HW79] or partitioning around medoids [PAM, KR90], which use only one data point at a time. All of these can be (more or less) easily modified to incorporate group constraints.

Let $G_N = \{g_1, \dots, g_N\}$, $g_n \in \{1, \dots, M\}$, denote a given prespecified classification of the N data points into $M \leq N$ groups G_m:

$$G_m = \{\mathbf{x}_n \in X_N | g_n = m\}, \quad m = 1, \dots, M.$$

For notational simplicity we will assume that the groups are disjoint in the following. Several parts could be easily extended to the case of overlapping groups with $g_n \subseteq \{1, \dots, M\}$, see the discussion in Section 4. Following [WCRS01] we define a constraint α_m for each group $m = 1, \dots, M$ which can take two possible values:

$$\alpha_m = \begin{cases} \text{must-link :} & \text{all group members must be in the same cluster} \\ \text{cannot-link :} & \text{all group members must be in different clusters} \end{cases}$$

Data points without any grouping constraints are put into singular dummy groups of size one with a must-link constraint. Obviously, groups with cannot-link constraint must not have more than K members: $|G_m| \leq K$.

[3] The machine learning community seems to prefer "online" algorithms of the "competitive learning" type, see [Rip96] for an overview.

2.1 Exchange Algorithms with Group Constraints

The cluster algorithm described in [HW79] as well as PAM by [KR90] follow the same basic principle: Both algorithms start with an initial partitioning of the data set, e.g., by using K random data points as initial centroids. In each iteration a single data point is selected and if a swap of cluster membership decreases the overall objective function D then this swap is done, otherwise the point remains in its cluster. The main difference between the two algorithms is that the first uses arbitrary points in space as cluster centroids, while PAM only allows data points in X_N as centroids (and calls them medoids).

If group constraints are to be fulfilled the following algorithm can be used:

1. Randomly select a subset of K different data points from X_N and use them as initial set of centroids C_K. Assign each data point $\mathbf{x}_n$ to the cluster of the closest centroid $c(\mathbf{x}_n)$, possibly violating the group constraints.
2. Select a group G_m by iterating through a random permutation of the numbers $1, \ldots, M$.
3. Depending on the type of group constraint α_m do one of the following:
   $\alpha_m =$ must-link: Assign all points $\mathbf{x} \in G_m$ together to each of the K clusters, recompute the system of centroids C_K and D for each assignment in turn and keep the optimal one.
   $\alpha_m =$ cannot-link: Find the optimal assignment of the $|G_m|$ points $\mathbf{x} \in G_m$ to the K clusters, recompute centroids C_K and D. Keep the new cluster assignment of the points
   - if the group did previously not fulfill the cannot-link constraint (first pass through the data), or
   - if the new assignment decreases D compared with a valid assignment from the previous iteration.
4. Remove empty clusters and repeat from step 2 until convergence or a prespecified maximum number of iterations.

Technical details for efficient computation of possible swaps in step 3 are given in the references cited above, an algorithm for optimal assignment of points with cannot-link constraint is shown in Section 2.3. After the first pass through the complete data set the following holds:

- all group constraints are fulfilled
- the algorithm accepts only swaps decreasing the overall performance measure D.

Because only a finite number of possible assignments of N data points to K clusters is possible and the algorithm is strictly decreasing in D, convergence is guaranteed. Note that convergence is only to a local minimum, in practice it is advisable to try several random initializations and use the best solution.

2.2 *K*-means with Group Constraints

Let us now consider a modification of the generalized K-means algorithm. The main difference to exchange algorithms is that centroids and cluster assignments are not recomputed after considering a single group of points, but always for all groups simultaneously. This usually increases the probability of getting stuck in a local minimum of D, but can be more efficiently implemented in highly vectorized interpreted languages like R.

1. Randomly select a subset of K different data points from X_N and use them as initial set of centroids C_K. Assign each data point $\mathbf{x}_n$ to the cluster of the closest centroid $c(\mathbf{x}_n)$.
2. Update the set of centroids holding the cluster memberships fixed.
3. For each group G_m, $m = 1, \ldots, M$, do one of the following depending on the type of group constraint α_m:
   $\alpha_m =$ must-link: Assign all points $\mathbf{x} \in G_m$ simultaneously to the cluster where the centroid has the minimum sum of distances to all group members.
   $\alpha_m =$ cannot-link: Find the optimal assignment of the $|G_m|$ points $\mathbf{x} \in G_m$ to the K clusters such that sum of distances of each group member to its respective cluster centroid is minimal.
4. Repeat from step 2 until convergence or a pre-specified maximum number of iterations.

The algorithm is again strictly decreasing such that convergence is guaranteed. This algorithm is very similar to the one proposed by [WCRS01], however they use a greedy search for the assignment of groups with cannot-link constraint, while we show below how the optimal configuration for each group can be found in each iteration.

2.3 Implementation in Flexclust

We have implemented the generalized K-means algorithm with group constraints as part of function `kcca()` in R package `flexclust` [Lei06]. Flexclust offers an extensible toolbox for K-centroids cluster analysis where users can easily combine several cluster algorithms with self-written and hence arbitrary distance functions. Following this basic design principle of extensibility users can specify a function implementing the group constraints as part of `"kccaFamily"` objects. This functions basically must encode the assignments of points to clusters from step 3 of the algorithm in Section 2.2. It takes a vector of old cluster assignments $c(\mathbf{x}_n)$, group memberships g_n and an $N \times K$ matrix of distances from each data point to each centroid as input, and returns a vector of length N of new cluster assignments[4]. The straightforward function `minSumClusters(cluster, group, distmat)` implements must-link constraints as described above.

Function `differentClusters(cluster, group, distmat)` implements cannot-link constraints. For each group G_m we have to solve a linear sum assignment problem. The optimal solution can be found in polynomial time of order $\mathcal{O}(K^3)$ using the so-called Hungarian method [e.g., PS82]. R package `clue` provides an implementation for $K \times K$ linear sum assignment problems in function `solve_LSAP()` [HB05], we have extended this to the $L \times K$ case by filling up non-square matrices with $K - L$ dummy rows where all entries are (basically) infinite.

There currently is no function implementing both must-link and cannot-link constraints simultaneously. However, this is not for technical reasons, but only because we had no application for it yet. All that needs to be done is combining the code of the two existing functions into one.

[4] The actual implementation in `flexclust` is a little bit more advanced: After convergence the function also computes the second-best cluster assignment of each group for visualization purposes.

3 Application Examples

3.1 Must-Link: Clustering Household Shopping Baskets

The application that first motivated us to research clustering with group constraints is market segmentation of shopping basket data, see [Bal05] for a comprehensive analysis of both artificial and real-world data. The latter are the “ZUMA subset” of the GfK Nürnberg household panel data [Pap01]: it consists of self-report data for shopping baskets of 40000 households over one year, on average 100 baskets per household. The data are binary indicators for 65 product groups like milk, cheese, washing powder or pet food and specify whether a product from the respective group has been bought or not. Clustering the complete data set at once was not possible due to the size of the data set, so a sampling scheme similar to the CLARA algorithm [KR90] was adopted to iteratively cluster parts of the data.

The task was the assignment of each household to one of several market segments and find product groups which are bought together. We tried several grouping strategies:

1. Summarize the data for each household, i.e., first compute a new data set where each observation corresponds to the sum of all shopping trips of each household for the complete year.
2. Cluster the original basket data without any group constraints, and then analyze the cross-tabulation of households and clusters.
3. Cluster the original basket data with a must-link constraint for each household.

1 and 2 did not give satisfactory market segments, probably due to the following reasons: The data matrix itself is very sparse, i.e., each basket contains only a few product groups which makes the clustering result of strategy 2 rather unstable. However, the product groups themselves are very common, so most households bought many of them at some point in time over the year, and aggregation as in strategy 1 looses the information which products were in the same baskets. Clustering with must-link constraints seems to be the right compromise in between and yields satisfactory results with profiled cluster centroids. Detailed results have to be omitted in this paper due to space restrictions and can be found in [Bal05], a journal paper summarizing the main results is under preparation.

3.2 Cannot-Link: Bootstrapping Finite Mixture Models

In [GL04] bootstrap methods are proposed for model diagnostics in finite mixture models. However, the parametric bootstrap with random initialization of the EM algorithm for analysis of model identification leads to label switching, a problem which has already received some attention in Bayesian mixture modelling. It has been shown that imposing a suitable ordering constraint on one parameter is difficult because the choice of parameter is crucial [RG97], or because there might even not exist a parameter which determines a unique labelling of the parameters [Ste00] suggests to use a decision theoretic approach to deal with label switching where label-invariant distance measures are used.

We propose to cluster the component specific parameters in order to determine a suitable labelling of the components under a cannot-link constraint which prevents that components from a model fitted to the same bootstrap sample are assigned to the same cluster. This approach can be used to either cluster the a-posteriori probabilities or the fitted parameters with different distance measures, see [GL05] for details.

This clustering strategy is applied to a simple artificial example of a Gaussian mixture regression model with 3 components where no suitable ordering constraint exists as each parameter overlaps at least for two components. In Figure 1 a sample of 50 individuals with observations for both x_2 values where the mixture component membership is fixed is given on the left. A model is fitted to the sample using the best of 5 repetitions of the EM algorithm. 100 parametric bootstrap samples are drawn from the fitted model and to each bootstrap sample a model is fitted using the best of 5 repetitions of the EM algorithm. The K-means clustering result of the standardized parameters of the bootstrap models is visualized using parallel coordinate plots on the right.

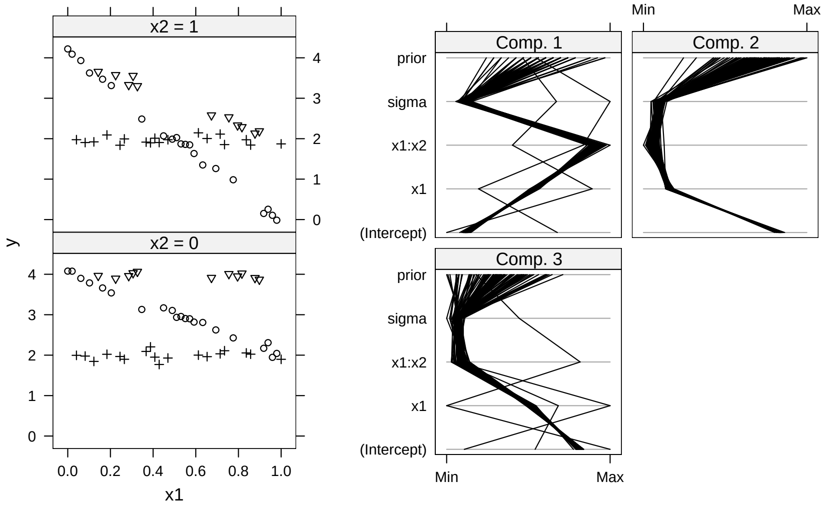

Fig. 1. Sample from the finite mixture (left) and parallel coordinate plot of the parameters of the 100 bootstrap models where the labelling is determined using K-means clustering under the cannot-link constraint (right).

4 Summary and Future Work

Using prior group information in clustering procedures is a natural task that has received surprisingly little attention in the literature so far. We have shown general solutions for two popular classes of cluster algorithms and two types of group constraints. These improve on existing solutions by replacing greedy cluster assignments by locally optimal ones in each iteration. An extensible software implementation in

R is available that allows users to easily program new types of constraints and hook them into existing clustering procedures. Implementation of exchange algorithms with group constraints is currently under investigation. A natural extension are overlapping groups with different types of constraints, which need to be checked for admissibility such that it cannot happen that two points are simultaneously in groups with must-link and cannot-link constraints. A possible way of clustering with multiple groupings is to represent those by several group vectors (which contain only non-overlapping groups) and applying the procedure presented above iteratively to each, but the details have yet to be worked out.

Another issue are other types of constraints or algorithmic solutions. E.g., we also have code for majority vote for must-link constraints where all points of a group are assigned to the cluster of the centroid that is closest to the majority of group members. This assignment is more robust, because it is not possible that a single outlier moves a complete group of points to a different cluster. The disadvantage is that convergence cannot be guaranteed because the algorithm is no longer strictly decreasing, although empirical evidence suggests that this seems not to be a problem in praxis.

Acknowledgement. This research was supported by the Austrian Academy of Sciences (ÖAW) through a DOC-FFORTE scholarship and the Austrian Science Foundation (FWF) under grant P17382.

References

[Bal05] Kerstin Balka. Neue Methoden zur Marktsegmentierung durch Clustern von gruppierten Warenkorbdaten. Master's thesis, Technische Universität Wien, Vienna, Austria, 2005. Friedrich Leisch, advisor.

[GL04] Bettina Grün and Friedrich Leisch. Bootstrapping finite mixture models. In Jaromir Antoch, editor, *Compstat 2004 — Proceedings in Computational Statistics*, pages 1115–1122. Physica Verlag, Heidelberg, 2004. ISBN 3-7908-1554-3.

[GL05] Bettina Grün and Friedrich Leisch. Detecting genuine multimodality in finite mixture models. Department of Statistics, Vienna University of Technology, Austria, *Submitted*, 2005.

[Grü02] Bettina Grün. Identifizierbarkeit von multinomialen Mischmodellen. Master's thesis, Technische Universität Wien, Vienna, Austria, 2002. Kurt Hornik and Friedrich Leisch, advisors.

[HB05] Kurt Hornik and Walter Boehm. *clue: Cluster Ensembles*, 2005. R package version 0.2-0.

[HW79] J. A. Hartigan and M. A. Wong. Algorithm AS136: A k-means clustering algorithm. *Applied Statistics*, 28(1):100–108, 1979.

[KR90] Leonard Kaufman and Peter J. Rousseeuw. *Finding Groups in Data.* John Wiley & Sons, Inc., New York, USA, 1990.

[Lei04] Friedrich Leisch. FlexMix: A general framework for finite mixture models and latent class regression in R. *Journal of Statistical Software*, 11(8):1–18, 2004.

[Lei06] Friedrich Leisch. A toolbox for k-centroids cluster analysis. *Computational Statistics and Data Analysis*, 2006. Accepted for publication.

[Mac67] J. MacQueen. Some methods for classification and analysis of multivariate observations. In L. M. Le Cam and J. Neyman, editors, *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, pages 281–297. University of California Press, Berkeley, CA, USA, 1967.

[MP00] Geoffrey J. McLachlan and David Peel. *Finite Mixture Models.* Wiley, 2000.

[Pap01] Georgios Papastefanou. The ZUMA scientific use file of the GfK ConsumerScan household panel 1995. In Georgios Papastefanou et al., editors, *Social and Economic Analyses with Consumer Panel Data*, pages 206–212. ZUMA Mannheim, 2001.

[PS82] C. Papadimitriou and K. Steiglitz. *Combinatorial Optimization: Algorithms and Complexity.* Prentice Hall, Englewood Cliffs, USA, 1982.

[R D05] R Development Core Team. *R: A language and environment for statistical computing.* R Foundation for Statistical Computing, Vienna, Austria, 2005. ISBN 3-900051-07-0.

[RG97] Sylvia Richardson and Peter J. Green. On Bayesian analysis of mixtures with an unknown number of components. *Journal of the Royal Statistical Society B*, 59(4):731–92, 1997.

[Rip96] Brian D. Ripley. *Pattern recognition and neural networks.* Cambridge University Press, Cambridge, UK, 1996.

[Ste00] Matthew Stephens. Dealing with label switching in mixture models. *Journal of the Royal Statistical Society B*, 62(4):795–809, 2000.

[WCRS01] Kiri Wagstaff, Claire Cardie, Seth Rogers, and Stefan Schroedl. Constrained k-means clustering with background knowledge. In *Proceedings of the Eighteenth International Conference on Machine Learning*, pages 577–584, 2001.