Ensemble Methods for Cluster Analysis

Kurt Hornik and Friedrich Leisch

1 Introduction

Ensemble methods create solutions to learning problems by constructing a set of individual (different) solutions ("base learners"), and subsequently suitably aggregating these, e.g., by weighted averaging of the predictions in regression, or by taking a weighted vote on the predictions in classification. Such methods, which include Bayesian model averaging (Hoeting et al., 1999), bagging (Breiman, 1996) and boosting (Friedman et al., 2000) have already become very popular for supervised learning problems (Dietterich, 2002).

Employing ensemble methods for cluster analysis can be attractive or even necessary for several reasons, the main three being as follows (see e.g. Strehl and Ghosh, 2002):

- To improve quality and robustness of the results. In general, aggregation yields algorithms with "low variance" in the statistical learning sense so that the results obtained by aggregation are more "structurally stable". For example, many clustering algorithms are sensitive to random initializations, choice of hyperparameters, or the order of data presentation in on-line learning scenarios. An obvious idea for possibly eliminating such *algorithmic* variability is to construct an ensemble with (randomly) varied characteristics of the base algorithm. This idea of "sampling from the algorithm" is used e.g. in the voting and voting/merging approaches of Dimitriadou et al. (2002, 2001), see also Section 3. Another idea is to try to improve quality via varying the data by resampling or reweighting. "Bagged Clustering" (Leisch, 1999), see also Section 2, constructs bootstrap samples; a similar approach is used in Dudoit and Fridlyand (2002). Other possible strategies include varying the "features" used for clustering (e.g., using various preprocessing schemes), and constructing "meta-clusterers" which combine the results of the application of different base algorithms as an attempt to reduce dependency of results on specific methods.

- To reuse existing knowledge. In applications, it may be desired to reuse legacy clusterings in order to improve or combine these. Typically, in such situations only the cluster labels are available, but not the original features or algorithms.

- To accommodate the needs of distributed computing. In many applications, it is not possible to use all data simultaneously. Data may not necessarily be available in a single location, or computational resources may be insufficient to use a base clusterer on the whole data set. More generally, clusterers can have access to either a subset of the objects ("object-distributed clustering") or the features ("feature-distributed clustering").

To fix notations and terminology, suppose we are given a set $\mathcal{X}$ of n objects, each holding the measurements on the same variables or features. A K-*clustering* of $\mathcal{X}$ assigns to each x_i in $\mathcal{X}$ a (sub-)probability K-vector $C(x_i) = (\mu_{i1}, \dots, \mu_{iK})$ (the "membership vector" of the object) with $\mu_{i1}, \dots, \mu_{iK} \geq 0$, $\sum_k \mu_{ik} \leq 1$. Formally,

$$C : \mathcal{X} \to M \in \mathbb{R}^{n \times K}; \qquad M \geq 0, \quad M1 \leq 1.$$

This framework includes both "crisp" (where each $C(x_i)$ is a unit vector) and fuzzy clustering, as well as incomplete (e.g., completely missing) results where $\sum_k \mu_{ik} < 1$. Changing the labels (which correspond to the columns of the membership matrix M) amounts to replacing M by $M\Pi$, where Π is a suitable permutation matrix. Finally, a *clusterer* is an algorithm producing a clustering.

Given an ensemble of clusterings, the following key distinction can be made for possible aggregation strategies to determine a "consensus" clustering. If each clustering is of the vector quantization type, aggregation can be based on the underlying prototypes (provided that these are available). Otherwise, if only the memberships are available, aggregation can proceed by finding a suitable clustering which "optimally represents" the base clusterings. "Bagged Clustering" and "Voting" are two very promising examples from these two aggregation categories, and will be described in more detail in Sections 2 and 3, respectively.

2 Aggregation Based on Prototypes

A large number of partitioning algorithms represent clusters by one prototype c_k for each cluster and (for crisp partitions) assign each observation x_i to the cluster of the closest prototype. Hence, the cluster memberships μ_{ik} can be written as

$$\mu_{ij} = \begin{cases} 1, & \Delta(x_i, c_j) = \min_k \Delta(x_i, c_k) \\ 0, & \text{otherwise} \end{cases}$$

for a suitable distance measure Δ. Fuzzy partitions assign memberships inversely proportional to distance (or ranks of distance). The well known K-means algorithm uses Euclidean distance as Δ and cluster means as prototypes.

If we are given B clusterings of the same set of objects $\mathcal{X}$ with K prototypes each, we may view the set of $B \times K$ prototypes c_{bk} as a new data set that can be used to assess the structural stability of the clusterer. Prototypes that show up often indicate "typical" clusters, while rare prototypes may indicate random fluctuations.

The bagged clustering algorithm (Leisch, 1999) uses this approach to find structurally more stable partitions: By repeatedly training on new data sets one gets different solutions which should on average be independent from training set influence and random initializations. A collection of B training sets can be obtained by sampling from the empirical distribution of the original data $\mathcal{X}$, i.e., by bootstrapping (Efron and Tibshirani, 1993).

The complete algorithm works as follows:

1. Construct B bootstrap training samples $\mathcal{X}_1, \dots, \mathcal{X}_B$ by drawing with replacement from the original sample $\mathcal{X}$.

2. Run the base clusterer (K-means, competitive learning, …) on each set, resulting in $B \times K$ prototypes $c_{11}, c_{12}, \ldots, c_{1K}, c_{21}, \ldots, c_{BK}$ where K is the number of prototypes used in the base method and c_{ij} is the j-th prototype found using $\mathcal{X}_i$.
3. Combine all prototypes into a new data set $\mathcal{C} = \{c_{11}, \ldots, c_{BK}\}$.
4. Run a hierarchical cluster algorithm on $\mathcal{C}$, resulting in the usual dendrogram.
5. Let $c(x) \in \mathcal{C}$ denote the prototype closest to x (minimum distance Δ). A partition of the original data can be obtained by cutting the dendrogram at a certain level, resulting in a partition $\mathcal{C}_B^1, \ldots, \mathcal{C}_B^m$, $1 \leq m \leq BK$, of set $\mathcal{C}$. Each point $x_i \in \mathcal{X}$ is now assigned to the cluster containing $c(x_i)$.

Bagged clustering combines the prototypes using hierarchical clustering because different data structures (convex, not convex, …) can be accounted for using different linkage methods and the resulting dendrograms can be easily interpreted by practitioners. But many other techniques could be used instead. Unfortunately it is not possible to compare prototypes of partitions directly (e.g., c_{11} with c_{21} etc.) due to the relabelling problem, see also Section 3 below for a more detailed discussion of this most important problem for all cluster ensemble methods.

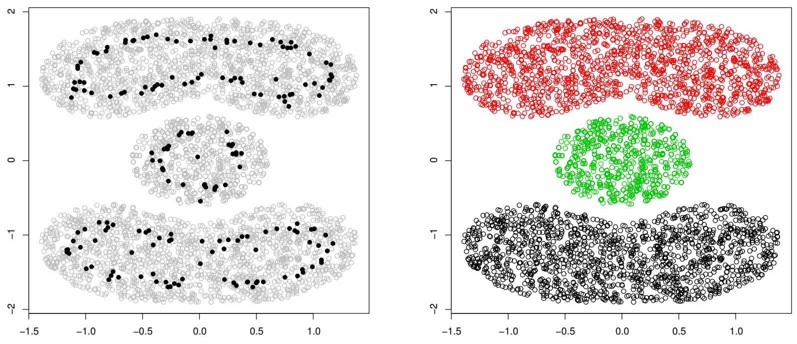

Figure 1: Cassini problem: 200 centers placed by bagged clustering (left) and final solution (right) by combining the 200 centers using hierarchical clustering.

Throughout this paper we use a 2-dimensional artificial data set called "Cassini" to demonstrate some aspects of cluster ensembles. The data set has 3900 objects in 3 groups (see Figure 1): 900 in the interior, 1500 each in the outer groups, all drawn uniformly from the respective shapes. The problem is "hard" for VQ-type base clusterers due to non-convexity of the outer groups, e.g., for the K-means algorithm with $K = 3$ the correct solution is a local minimum only, the global minimum of the K-means objective function splits one of the outer groups into two halves and ignores the inner group.

We apply the bagged cluster algorithm to this data using $B = 10$ bootstrap training samples and K-means as base method with $K = 20$ centers in each run. The

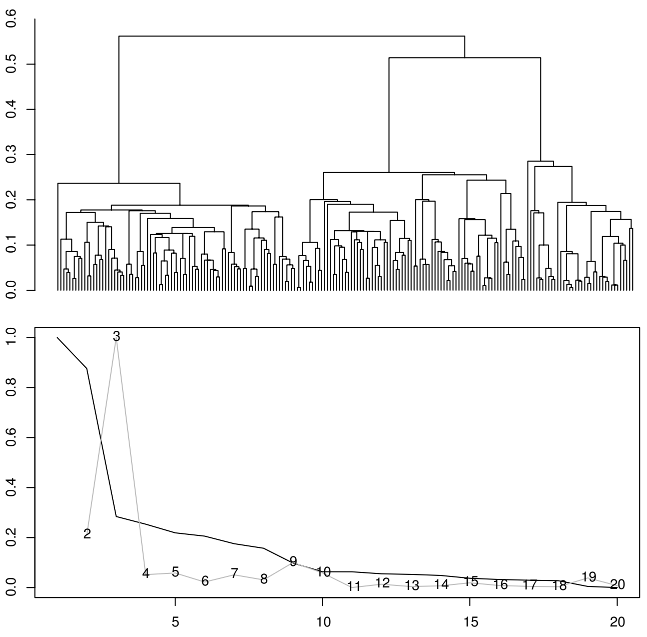

Figure 2: Cassini problem: Hierarchical clustering of 200 bagged cluster centers using single linkage. The upper plot shows the usual dendrogram. The lower plot shows the height of the splits together with their first differences, which can be used to determine the number of clusters (here 3).

left plot in Figure 1 shows the resulting 200 centers. We then perform hierarchical clustering (Euclidean distance, single linkage) on these 200 points, see Figure 2. The three-cluster partition which results from cutting the dendrogram into its three main branches can be seen in the right plot in Figure 1. It recovers the three clusters without error. Note that direct hierarchical clustering of this data set is infeasible due to its size.

3 Aggregation Based on Memberships

If aggregation is to be based on cluster memberships $M_1, \dots, M_B$ only, a natural way to proceed is by looking for clusterings which "optimally represent" the ensemble. (Note that due to possible relabeling we cannot simply compute average memberships.) Suppose that $d(M, \tilde{M})$ measures dissimilarity (or distance) between two clusterings C and $\tilde{C}$ with corresponding membership matrices M and $\tilde{M}$, respectively. Given d, we can for example look for Ms which minimize average dissimilarity, i.e., which solve

$$M^* = \operatorname*{argmin}_{M \in \mathcal{M}} \sum_{b=1}^{B} d(M, M_b)$$

over a suitable set $\mathcal{M}$ of membership matrices M. If $\mathcal{M}$ contains all crisp clusterings, Gordon and Vichi (1998) call M^* the *median partition*, if $\mathcal{M} = \{M_1, \ldots, M_B\}$ it is the *medoid partition*.

We refer to this minimization problem as the (simple) *cluster ensemble problem*. Many extensions are possible, such as minimizing $\sum_b \omega_b d(M, M_b) + \lambda\Phi(M)$, where the ω_b are weights quantifying "importance", and Φ can e.g. measure fuzziness, thus converting the above hard-constrained simple problem into a soft-constrained extended one. Also, one could consider criterion functions resulting in yet more robust solutions, such as the median or trimmed mean of the distances $d(M, M_b)$.

Unfortunately, the simple cluster ensemble problem is computationally very hard. Even if "only" crisp solutions are sought, it would in general be necessary to search all possible crisp clusterings (the number of which is of the order $(K+1)^n$) for the optimum. Such exhaustive search is clearly impossible for most applications. Local strategies, e.g. by repeating random reassigning until no further improvement is obtained, or Boltzmann-machine type extensions (Strehl and Ghosh, 2002) are still expensive and not guaranteed to find the global optimum.

Gordon and Vichi (1998) use the Rand index (Rand, 1971) as distance measure d, while Krieger and Green (1999) use the Rand index corrected for agreement by chance (Hubert and Arabie, 1985). Solving for M^* is NP-hard in both cases, hence the corresponding mathematical programming problems scale bad in the number of observations. Krieger and Green (1999) propose a greedy search algorithm together with "smart" initialization.

The situation can considerably be improved if more information on the structure of the optimal clustering is available. Dimitriadou et al. (2002) use the distance measure

$$d_{\mathrm{DWH}}(M, \tilde{M}) = \min_{\Pi} \|M - \tilde{M}\Pi\|^2$$

where the minimum is taken over all permutation matrices Π. In the crisp case, d_{DWH} counts (a multiple of) the number of differently labeled objects after optimal relabeling. For this distance measure, one can show that the optimal (fuzzy) solution M^* to the cluster ensemble problem is of the form

$$M = \frac{1}{B} \sum\nolimits_{b=1}^{B} M_b \Pi_b$$

for suitable permutation matrices $\Pi_1, \ldots, \Pi_B$. In the all-crisp case, the aggregated memberships are obtained by simple majority voting after relabeling, which motivates the name "voting" for the proposed framework. Simultaneous determination of the permutation matrices still being computationally hard, the above representation motivates a greedy forward aggregation algorithm where in each step b, a locally optimal Π_b^* for relabeling is determined, and the optimal aggregation M_b^* of $M_1\Pi_1^*, \ldots, M_b\Pi_b^*$ is obtained by on-line averaging. The locally optimal permutation matrix can be determined via linear programming using the so-called Hungarian method for solving the weighted bi-partite graph matching problem (e.g., Papadimitiou and Steiglitz (1982)).

Formally,

$$\begin{aligned} \Pi_b^* &= \underset{\Pi}{\operatorname{argmin}}\, d_{\mathrm{DWH}}(M_{b-1}^*, M_b\Pi) \\ M_b^* &= (1 - 1/b)M_{b-1}^* + (1/b)M_b\Pi_b^* \end{aligned}$$

The final M_B^* is the consensus clustering obtained by "voting".

Figure 3 shows the improvements on the Cassini problem obtained by successive voting using K-means with $K = 3$ clusters as the base learner. Figure 4 demonstrates

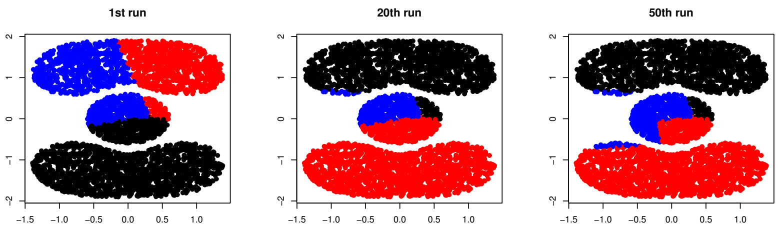

Figure 3: Aggregation by voting on k-means

how voting on the aggregated results of $B = 50$ runs of K-means, Hard Competitive Learning, and an on-line version of fuzzy C-means further improves performance, resulting in (almost) perfect learning of the underlying structure.

4 Summary and Outlook

The main focus of this paper is on aggregation strategies for cluster ensembles. It may be desirable to subject the thus obtained consensus clustering to further computations, such as for collapsing labels representing similar groups, using for example the "Merging" procedure in Dimitriadou et al. (2001). Cluster ensembles can also be used for tuning hyper-parameters of clustering algorithms, such as determining the number of clusters to be employed (Dudoit and Fridlyand, 2002).

Cluster ensembles have already been successfully employed in a wide range of application domains, including market segmentation (Dolničar and Leisch, 2003) and the analysis of fMRI data (Barth et al., 2003). Nevertheless, there is still room for substantial improvements of the underlying theory. For example, it is currently not known under which conditions to solutions to the (unconstrained) cluster ensemble problem can be represented as convex combinations ("weighted voting") of the possibly relabeled membership matrices (including the result of Dimitriadou et al. (2002) as a special case), or can be computed in polynomial time. Such knowledge could result in the construction of substantially more efficient aggregation algorithms, making large-scale application problems computationally tractable.

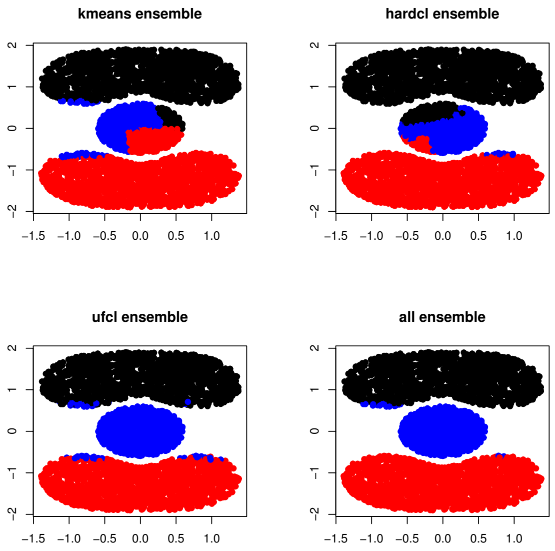

Figure 4: Aggregation by voting on voting on k-means, Hard Competitive Learning, and fuzzy c-means

Bibliography

Barth, M., Dimitriadou, E., Hornik, K., and Moser, E. (2003). Ensemble clustering of fMRI data. In *20th Annual Meeting of the European Society for Magnetic Resonance in Medicine and Biology (ESMRMB 2003)*.

Breiman, L. (1996). Bagging predictors. *Machine Learning*, 24:123–140.

Dietterich, T. G. (2002). Ensemble learning. In Arbib, M. A., editor, *The Handbook of Brain Theory and Neural Networks*, pages 405–408. MIT Press, Cambridge, Mass.

Dimitriadou, E., Weingessel, A., and Hornik, K. (2001). Voting-merging: An ensemble method for clustering. In Dorffner, G., Bischof, H., and Hornik, K., editors, *Artificial Neural Networks – ICANN 2001*, volume 2130 of *LNCS*, pages 217–224. Springer, Berlin.

Dimitriadou, E., Weingessel, A., and Hornik, K. (2002). A combination scheme for fuzzy clustering. *International Journal of Pattern Recognition and Artificial Intelligence*, 16:901–912.

Dolničar, S. and Leisch, F. (2003). Winter tourist segments in Austria: Identifying stable vacation styles using bagged clustering techniques. *Journal of Travel Research*, 41:281–292.

Dudoit, S. and Fridlyand, J. (2002). A prediction-based resampling method to estimate the number of clusters in a dataset. *Genome Biology*, 3:0036.1–0036.21.

Efron, B. and Tibshirani, R. J. (1993). *An introduction to the bootstrap*. Monographs on Statistics and Applied Probability. Chapman & Hall, New York.

Friedman, J., Hastie, T., and Tibshirani, R. (2000). Additive logistic regression: A statistical view of boosting. *The Annals of Statistics*, 28:337–407.

Gordon, A. D. and Vichi, M. (1998). Partitions of partitions. *Journal of Classification*, 15:265–285.

Hoeting, J., Madigan, D., Raftery, A., and Volinsky, C. (1999). Bayesian model averaging: A tutorial. *Statistical Science*, 14:382–401.

Hubert, L. and Arabie, P. (1985). Comparing partitions. *Journal of Classification*, 2:193–218.

Krieger, A. M. and Green, P. E. (1999). A generalized Rand-index method for consensus clustering of separate partitions of the same data base. *Journal of Classification*, 16:63–89.

Leisch, F. (1999). Bagged clustering. Working paper, SFB “Adaptive Information Systems and Modeling in Economics and Management Science”, Vienna University of Economics and Business Administration.

Papadimitiou, C. and Steiglitz, K. (1982). *Combinatorial Optimization: Algorithms and Complexity*. Prentice Hall, Englewood Cliffs.

Rand, W. M. (1971). Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 66:846–850.

Strehl, A. and Ghosh, J. (2002). Cluster ensembles – a knowledge reuse framework for combining multiple partitions. *Journal on Machine Learning Research*, 3:583–617.