Cluster Ensembles

Kurt Hornik

Ensemble Methods

Solve learning problem by

- constructing a set of individual (different) solutions (using "base learners");
- suitably aggregating these.

(E.g., weighted averaging of or voting on the predictions).

Includes Bayesian model averaging, bagging, and boosting.

Popular for supervised learning problems (regression and classification).

Why Cluster Ensembles?

- Improve quality and robustness of results
  - (Random) variation of characteristics of one clustering algorithm (starting values, hyperparameters, order of presentation for on-line algorithms): e.g., "Cluster Voting" (Dimitriadou, Weingessel & Hornik, 2002).
  - (Random) variation of data (resample or reweight): e.g., "Bagged Clustering" (Leisch, 1999), also used in Dudoit & Fridlyand (2002).
  - Use of different features to represent the objects (e.g., raw versus pre-processed images)
  - Application of several different base clusterers

Why Cluster Ensembles?

- Knowledge Reuse: Improve or combine available legacy clusterings, typically without going back to the original features or algorithms
- Distributed Computing: E.g., if data is not necessarily available in a single centralized location
  - Feature-distributed clustering: each clusterer has only access to a subset of the features
  - Object-distributed clustering: each clusterer has only access to a subset of the objects

Notations and Terminology

Given a set $\mathcal{X}$ of n objects, a K-clustering of $\mathcal{X}$ assigns to each x_i in $\mathcal{X}$ a (sub-)probability K-vector $C(x_i) = (\mu_{i1}, \ldots, \mu_{iK})$ (the "membership vector" of the object) with $\mu_{i1}, \ldots, \mu_{iK} \geq 0$, $\sum_k \mu_{ik} \leq 1$. I.e.,

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

Above deals with fuzzy clustering and "missing" labels.

Relabeling $\Leftrightarrow$ $M\Pi$, Π permutation matrix.

"Clusterers" are algorithms to produce clusterings.

A First Idea

Use the average pairwise co-labeling frequencies as distance between the objects, and recluster (Fred & Jain, 2002). I.e., use

$$D = HH'/B = (M_1 M_1' + \cdots + M_B M_B')/B$$

as the distance matrix, where

$$H = [M_1, \ldots, M_B]$$

is the "hypergraph" of the cluster ensemble (note: $[MM']_{ij} = \sum_k \mu_{ik}\mu_{jk}$).

Problem: $O(n^2)$.

Aggregation Based On Prototypes

Idea: if all clusterers in the ensemble are of the VQ type (and the prototypes are available): analyze the ensemble of prototypes.

I.e., assume that there is a distance measure Δ on the "feature space" and that for each base clusterer there are $c_1, \ldots, c_K$ such that

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

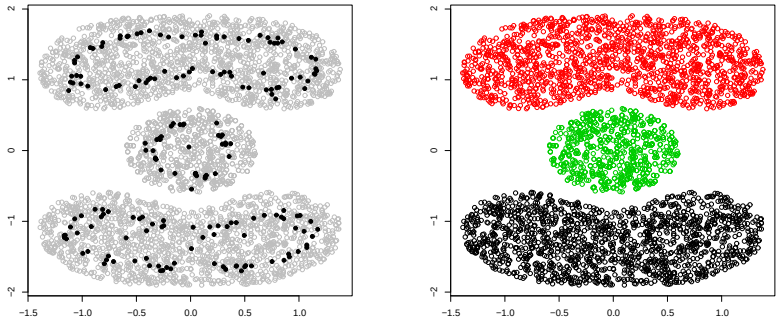
Bagged Clustering

- Construct B bootstrap samples $\mathcal{X}_1, \ldots, \mathcal{X}_B$ from $\mathcal{X}$.
- Run the base clusterer on each $\mathcal{X}_b$.
- Combine all prototypes into a new data set $\mathcal{C} = \{c_{11}, \ldots, c_{BK}\}$.
- Run a hierarchical cluster algorithm on $\mathcal{C}$.
- Let $c(x) \in \mathcal{C}$ be the prototype closest to x. Cut the dendrogram at a certain level, giving a partition $\mathcal{C}_1, \ldots, \mathcal{C}_m$ of $\mathcal{C}$. Assign each x to the cluster containing $c(x)$.

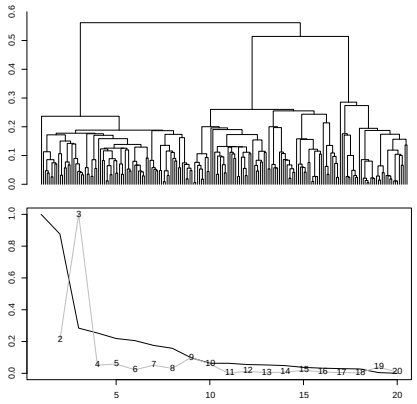
Example: Cassini

- 2-dimensional data set
- 3900 objects in 3 groups: 900 in the interior, 1500 each in the outer groups
- objects uniformly drawn from the respective groups
- "hard" for VQ-type base clusterers due to non-convexity

Cassini: Bagged Clustering



Cassini: Bagged Clustering



Aggregation Based on Memberships

What if "only" memberships are available or to be used?

Basic idea: find a "suitable" clustering M which "optimally represents" the ensemble of base clusterings $M_1, \ldots, M_B$.

Cluster Ensemble Problem

Define optimal aggregation by minimizing dissimilarity over a set of suitable clusterings. Simple problem:

$$\sum_{b=1}^{B} d(M, M_b) \rightarrow \text{min!}$$

with M suitably constrained (e.g., crisp); extended problem e.g.

$$\sum_{b=1}^{B} \omega_b d(M, M_b) + \lambda \Phi(M) \rightarrow \text{min!}$$

(could e.g. have ω and Φ measure importance and fuzziness, respectively).

Dissimilarity Measures

Many many many ...

In Dimitriadou, Weingessel & Hornik (2002):

$$d_{DWH}(M, \tilde{M}) = \min_{\Pi} \|M - \tilde{M}\Pi\|^2 = \min_{\Pi} \sum_{i,l} (\mu_{i,l} - \tilde{\mu}_{i,\pi(l)})^2$$

In the crisp case: number of discordant label pairs after optimal relabeling.

Dissimilarity Measures

In Strehl & Ghosh (2002):

$$d_{SG}(M, \tilde{M}) = -\frac{I(M, \tilde{M})}{\sqrt{H(M)}\sqrt{H(\tilde{M})}}$$

(negative normalized mutual information), where

$$I(M, \tilde{M}) = \sum_{l,\lambda} n_{l,\lambda} \log \frac{n_{l,\lambda} n_{\cdot,\cdot}}{n_{l,\cdot} n_{\cdot,\lambda}}$$

with $n_{l,\lambda} = \sum_i \mu_{i,l} \tilde{\mu}_{i,\lambda}$ is the l, λ element of the cross-labeling matrix $M' \tilde{M}$.

Rand Indices for Measuring Similarity

	Second Partition		
First Partition	N of Pairs in Same	N of Pairs in Diff	Sum
N of Pairs in Same Cluster	A	B	$A+B$
N of Pairs in Different Cluster	C	D	$C+D$
Sum	$A+C$	$B+D$	N

(crisp only, note $N = n(n-1)/2$).

Unadjusted Rand Index: $(A+D)/N$; adjusted one:

$$\frac{N(A+D) - ((A+B)(A+C) + (C+D)(B+D))}{N^2 - ((A+B)(A+C) + (C+D)(B+D))}$$

(cf. Cohen's Kappa).

Solving the Cluster Ensemble Problem

Hard even in the "simple" case.

Even if "only" crisp solutions are sought: order $(K+1)^n$ possible crisp clusterings.

Exhaustive search basically impossible.

"Greedy" local search:

repeat for each object try(relabel)
until no further improvement

(and perhaps modify a la Boltzmann machine): expensive and local.

Averaging Memberships After Relabeling

Dimitriadou, Weingessel & Hornik (2002) show that for $d = d_{DWH}$, the optimal (fuzzy) M is of the form

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

for suitable permutation matrices $\Pi_1, \ldots, \Pi_B$.

Motivates a greedy forward aggregation algorithm: in each step b, determine a locally optimal Π_b for relabeling, and average: i.e., given M^*_{b-1},

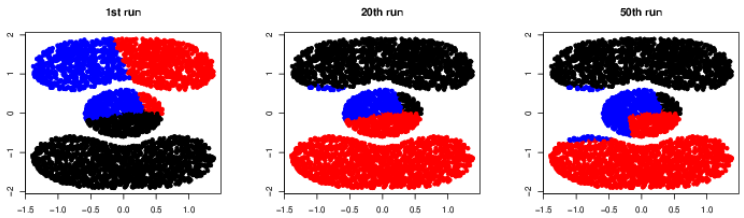
$$\Pi^*_b = \text{argmin}_\Pi d_{DWH}(M^*_{b-1}, M_b\Pi)$$
$$M^*_b = (1 - 1/b)M^*_{b-1} + (1/b)M_b\Pi^*_b$$

Averaging Memberships After Relabeling

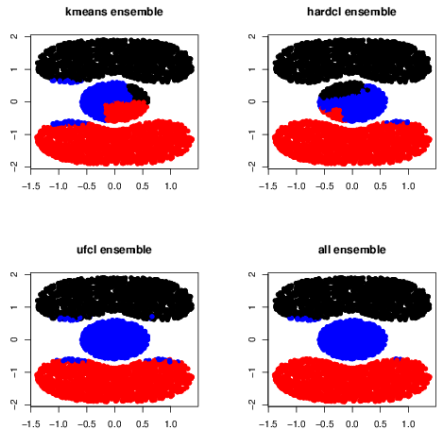
In (pseudo)code:

```
aggClMems <- memberships(baseClResults[[1]])
for(b in seq(along = baseClResults)[-1]) {
  newMems <- memberships(baseClResults[[b]]
  perm <- matchLabels(crossprod(aggClMems, newMems))
  aggClMems <- ((b - 1) * aggClMems + newMems[, perm]) / b
}
return(aggClMems)
```


Cassini: Voting



Cassini: Voting



More Aggregation Strategies

Consider again $H = [M_1, \ldots, M_B]$. In the crisp case, this is the "hypergraph" representation of the clusterings (objects are vertices, clusters are hyperedges).

- Can use hypergraph partition algorithms from machine learning (Strehl & Ghosh, 2002).
- Cluster the columns of H using the binary Jaccard coefficient (ratio of the intersection to the union of objects corresponding to the hyperedges), called the Meta-CLustering Algorithm (MCLA) in Strehl & Ghosh, 2002). Only works in the crisp case.

Benchmarking

Ah yes that would be nice . . .

Are there good benchmarking data sets?

Implementation

Bagged clustering and other tools available in R extension packages e1071 and cclust. Voting et al are "work in progress".

One problem: luck of class hierarchy for clusterings (but: if results have no class, can assume vectors are labelings and matrices are memberships).

Future Research

- When is the optimal fuzzy ensemble of the form

$$\sum_b \alpha_b M_b \Pi_b, \qquad \alpha_1, \ldots, \alpha_B \geq 0; \quad \sum_b \alpha_b \leq 1,$$

  i.e., a (sub-)convex combination of the re-labeled memberships?

- Matching of labels to optimize $\sum \omega_{i,\pi(i)}$ for given $[\omega_{ij}]$ can be done via linear programming (bipartite graph matching problems). Is there a generalization useful for simultaneously determining the optimal relabelings for "voting" a la DWH?

What We Really Should Do

What we have: collection of clustering objects $M_1, \ldots, M_B$.

Why only compute "averages"?

In addition, look at variability or even better, "structure" in these objects. I.e.,

Cluster clusterings!

If this is prototype-based, averaging strategies can be used for computing "centers".

Work in progress ...