Kernel-estimated Nonparametric Overlap-Based Syncytial Clustering

Israel A. Almodóvar-Rivera, PhD

Department of Biostatistics and Epidemiology
University of Puerto Rico
Medical Science Campus
Graduate School of Public Health
israel.almodovar@upr.edu

Introduction

- Commonly-used clustering algorithms usually find ellipsoidal, spherical or other regular-structured clusters, but are more challenged when the underlying groups lack formal structure or definition.
- Syncytial clustering is the name that we introduce for methods that merge groups obtained from standard clustering algorithms in order to reveal complex group structure in the data.

Methodology: Problem Setup

Let $\boldsymbol{\Xi} = \{\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_n\}$ be a random sample of n p-dimensional observations, with each

$$\boldsymbol{X}_i \sim \prod_{c=1}^{C} [f_c(\boldsymbol{x})]^{\zeta_{ic}}, \tag{1}$$

where C is the number of groups, $\zeta_{ic} = \mathcal{I}_{(\boldsymbol{X}_i \in \boldsymbol{\mathcal{C}}_c)}$ with $\mathcal{I}_{(\mathcal{Z})} = 1$ if $\mathcal{Z}$ holds and 0 otherwise, $f_c(\boldsymbol{x})$ is the cluster-specific density of an observation in the cth cluster and $\boldsymbol{\mathcal{C}}_c$ is the set of observations in the sample from that group.

Problem Set-up (cont'd)

We can model $\boldsymbol{X}_i \in \boldsymbol{\Xi}$ as $\boldsymbol{X}_i \sim \prod_{c=1}^{C} \prod_{k=1}^{k_c} [h(\|\boldsymbol{x} - \boldsymbol{\mu}_k^{\boldsymbol{c}_c}\|)]^{\zeta_{ik}^{\boldsymbol{c}_c}}$, or equivalently as

$$\boldsymbol{X}_i \sim \prod_{k=1}^{K} [h(\|\boldsymbol{x} - \boldsymbol{\mu}_k^{\circ}\|)]^{\zeta_{ik}^{\circ}}, \tag{2}$$

where ζ_{ik}° and $\boldsymbol{\mu}_k^{\circ}$ for $k = 1, 2, \ldots, K$ are renumerations, respectively, of all the $\zeta_{ik}^{\boldsymbol{c}_c}$ and $\boldsymbol{\mu}_k^{\boldsymbol{c}_c}$ for $k = 1, 2, \ldots, k_c, c = 1, 2, \ldots, C$. Therefore, $K = \sum_{c=1}^{C} k_c$, $\zeta_{ic} = \sum_{k=1}^{k_c} \zeta_{ik}^{\boldsymbol{c}_c}$ for $c = 1, 2, \ldots, C$ and $\sum_{k=1}^{K} \zeta_{ik}^{\circ} \equiv \sum_{c=1}^{C} \sum_{k=1}^{k_c} \zeta_{ik}^{\boldsymbol{c}_c} = 1$ (however, both K and C are also unknown).

Problem Set-up (cont'd)

From the $\hat{K}$-groups solution, define the ith residual ($i = 1, 2, \ldots, n$) as

$$\hat{\boldsymbol{\epsilon}}_i = \boldsymbol{X}_i - \sum_{k=1}^{\hat{K}} \hat{\boldsymbol{\mu}}_k^{\circ} \hat{\zeta}_{ik}^{\circ}; \tag{3}$$

where $\hat{\boldsymbol{\mu}}_k^{\circ}$ is the multivariate mean vector of the observations in the kth group and $\hat{\zeta}_{ik}^{\circ} = \mathcal{I}_{(\boldsymbol{X}_i \in\ k\text{th } k\text{-means group})}$. From (3), we obtain the normed residuals, that is, we obtain

$$\hat{\Psi}_i = \sqrt{\hat{\boldsymbol{\epsilon}}_i' \hat{\boldsymbol{\epsilon}}_i} = \| \boldsymbol{X}_i - \sum_{i=1}^{\hat{K}} \hat{\zeta}_{ik}^{\circ} \hat{\boldsymbol{\mu}}_k^{\circ} \| \tag{4}$$

for $i = 1, 2, \ldots, n; k = 1, 2, \ldots, \hat{K}$. These $\hat{\Psi}_1, \hat{\Psi}_2, \ldots \hat{\Psi}_n$ may be viewed as a random sample with density function $h_{\Psi}(\cdot)$ and CDF $H_{\Psi}(\cdot)$ and having support in $[0, \infty)$.

Pairwise overlap between groups

Maitra and Melnykov (2010) defined the pairwise overlap of two mixture components as the sum of the misclassification probabilities $\omega_{lk} \equiv \omega_{kl} = \omega_{l|k} + \omega_{k|l}$ with

$$\omega_{l|k} = \mathbb{P}[\boldsymbol{X} \text{ is assigned to } \boldsymbol{\mathcal{C}}_l \mid \boldsymbol{X} \text{ is truly in } \boldsymbol{\mathcal{C}}_k]. \tag{5}$$

Overlap between two groups is an indicator of the extent to which they are indistinguishable from each other.

Pairwise overlap between two k-means groups

The pairwise overlap (5) between two groups can generally be calculated from $H_\Psi(\cdot)$ as

$$\omega_{l|k} = \mathbb{P}\left(\|\boldsymbol{X} - \boldsymbol{\mu}_l\| < \|\boldsymbol{X} - \boldsymbol{\mu}_k\| \mid \boldsymbol{X} \in \boldsymbol{\mathcal{C}}_k\right) = 1 - \mathbb{P}\left(\Psi_k < \Psi_{l(k)}\right) \tag{6}$$

where Ψ_k represents the normed residual obtained from the kth group, and $\Psi_{l(k)}$ represents the normed *pseudo-residual* which we define as the norm of the remainder that is obtained by subtracting the lth cluster mean $\boldsymbol{\mu}_l$ from an observation $\boldsymbol{X} \in \boldsymbol{\mathcal{C}}_k$.

Pairwise overlap between two k-means groups (cont'd)

Calculation of $\mathbb{P}\left(\Psi_k < \Psi_{l(k)}\right)$ is not as straightforward. So we estimate $\mathbb{P}\left(\Psi_k < \Psi_{l(k)}\right)$ using a naïve average estimator

$$\hat{\mathbb{P}}\left(\Psi_k < \Psi_{l(k)}\right) = \frac{1}{n_k^\circ} \sum_{i=1}^{n} \hat{\zeta}_{ik}^\circ \hat{H}_\Psi(\|\boldsymbol{X}_i - \hat{\boldsymbol{\mu}}_l^\circ\|; \hat{b}), \tag{7}$$

where $n_k^\circ = \sum_{i=1}^{n} \hat{\zeta}_{ik}^\circ$, $\hat{H}_\Psi$ is a smooth estimator of the CDF and $\hat{b}$ is the smoothing parameter. Similar estimates of $\omega_{k|l}$, and therefore ω_{kl}, can be obtained.

Pairwise overlap between two composite groups

A composite group is one that can be further decomposed into sub-populations.

Let $\omega_{\boldsymbol{\mathcal{C}}_l|\boldsymbol{\mathcal{C}}_k}$ be defined as in (5) but for composite groups. That is, we use $\omega_{\boldsymbol{\mathcal{C}}_l|\boldsymbol{\mathcal{C}}_k}$ rather than $\omega_{l|k}$ in order to specify that the overlap measure is between composite clusters $\boldsymbol{\mathcal{C}}_l$ and $\boldsymbol{\mathcal{C}}_k$. Now $\omega_{\boldsymbol{\mathcal{C}}_l|\boldsymbol{\mathcal{C}}_k} = 1 - \mathbb{P}[\min_{r\in\boldsymbol{\mathcal{C}}_k} \|\boldsymbol{X} - \boldsymbol{\mu}_r\| < \min_{j\in\boldsymbol{\mathcal{C}}_l} \|\boldsymbol{X} - \boldsymbol{\mu}_j\| \mid \boldsymbol{X} \in \boldsymbol{\mathcal{C}}_k]$.

Pairwise overlap between two composite groups (cont'd)

Suppose now that $\mathcal{C}^{\circ}_{s\subset k}$ is the sth spherical sub-cluster of $\mathcal{C}_k$ with mean $\boldsymbol{\mu}^{\circ}_s$, $s = 1, 2, \ldots, |\mathcal{C}_k|$, with $|\mathcal{C}_k|$ being the number of spherical sub-clusters in $\mathcal{C}_k$. The density of $\boldsymbol{X}$ is defined through its (sth) sub-cluster and so

$$\begin{aligned}\mathbb{P}\left(\min_{r\in\mathcal{C}_k} \|\boldsymbol{X} - \boldsymbol{\mu}_r\| \leq y \mid \boldsymbol{X} \in \mathcal{C}_k\right) &= 1 - \mathbb{P}\left(\min\nolimits_{r\in\mathcal{C}_k} \Psi_r > y\right) \\ &= 1 - \left[1 - \mathbb{P}\left(\Psi_r \leq y\right)\right]^{|\mathcal{C}_k|} \quad (8)\end{aligned}$$

where Ψ_r is a normed residual (obtained, for instance, from the k-means solution) for the rth spherically-dispersed subgroup in the kth cluster.

Pairwise overlap between two composite groups (cont'd)

From (5), and using the same ideas as in (7) we get the naïve estimator

$$\hat{\omega}_{\mathcal{C}_l|\mathcal{C}_k} = \left[1 - \frac{1}{n_c}\sum_{i=1}^{n_c} \hat{\zeta}_{ic} \hat{H}_{\Psi}(\min_{r\in\mathcal{C}_l} \|\boldsymbol{X}_i - \boldsymbol{\mu}_r\|; \hat{b}),\right]^{|\mathcal{C}_k|} \tag{9}$$

where $\hat{H}_{\Psi}$ is a smooth estimator of the CDF, $\hat{b}$ is the smoothing parameter and similarly for $\hat{\omega}_{\mathcal{C}_k|\mathcal{C}_l}$, from where we calculate $\hat{\omega}_{\mathcal{C}_l\mathcal{C}_k} \equiv \hat{\omega}_{\mathcal{C}_k\mathcal{C}_l} = \hat{\omega}_{\mathcal{C}_l|\mathcal{C}_k} + \hat{\omega}_{\mathcal{C}_k|\mathcal{C}_l}$.

KNOB-SynC Algorithm

The initial overlap calculation phase

This phase starts with the output of the k-means phase.

1. For each observation $\boldsymbol{X}_i, i = 1, 2, \ldots, n$, compute its normed residual $\hat{\Psi}_i = \sqrt{\hat{\boldsymbol{\epsilon}}_i' \hat{\boldsymbol{\epsilon}}_i}$ where $\hat{\boldsymbol{\epsilon}}_i$ is defined as in (3). Also for each observation $\boldsymbol{X}_i$, obtain thenormed pseudo-residual $\hat{\Psi}_{i;l(k)} = \|\boldsymbol{X}_i - \hat{\boldsymbol{\mu}}_l\|$ for $\boldsymbol{X}_i \in \boldsymbol{\mathcal{C}}_k$, and $l \neq k \in \{1, 2, \ldots, \hat{K}\}$.
2. Using the set of normed residuals $\{\hat{\Psi}_i; i = 1, 2, \ldots, n\}$, obtain its kernel-estimated CDF.
3. For any two groups $k \neq l \in \{1, 2, \ldots, \hat{K}\}$, estimate the pairwise overlap $\hat{\omega}_{lk} = \hat{\omega}_{l|k} + \hat{\omega}_{k|l}$.
4. Obtain the estimated overlap matrix $\hat{\boldsymbol{\Omega}}^{(1)}$.
5. From the overlap matrix $\hat{\boldsymbol{\Omega}}^{(1)}$, calculate the generalized overlap $\ddot{\hat{\omega}}$. Call it $\ddot{\hat{\omega}}^{(1)}$.

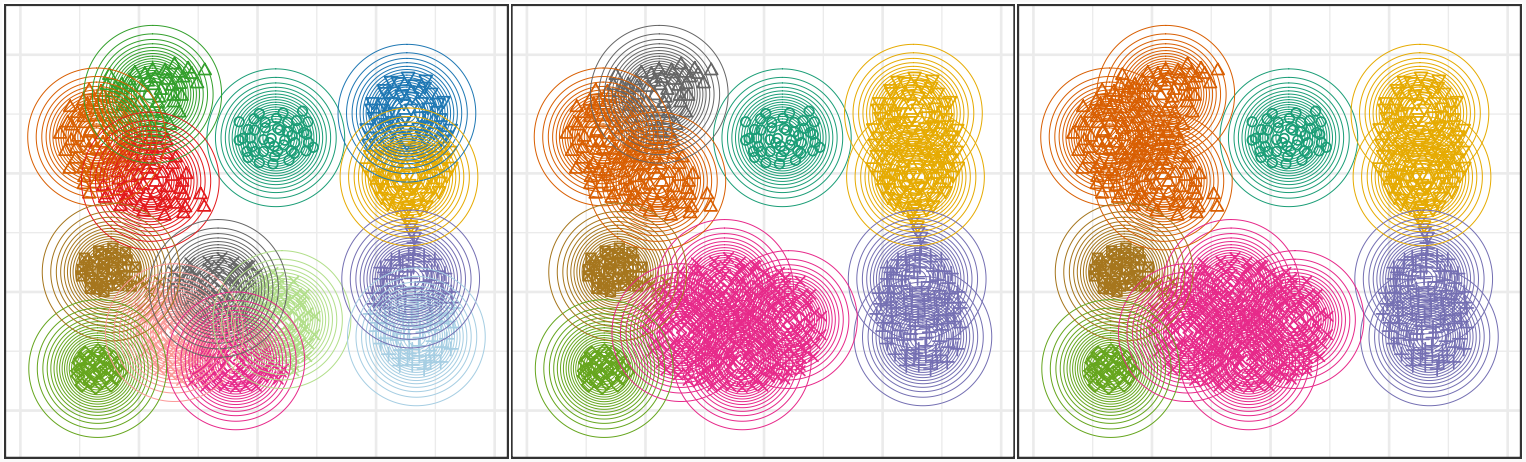
The merging phase

- The merging phase start only if $\ddot{\hat{\omega}}^{(1)} \not\geq 4\check{\hat{\omega}}$ or if $\ddot{\hat{\omega}}^{(1)} \not\approx 0$.
- This phase iteratively proceeds for $\ell = 1, 2, \ldots$ as per the following steps:
  1. Merge the groups with the maximum overlap and every pair of groups with individual pairwise overlaps substantially larger than the generalized overlap $\ddot{\hat{\omega}}^{(\ell)}$. That is, merge every pair of groups $\mathcal{C}_k$, $\mathcal{C}_l$, $k \neq l$ such that $\hat{\omega}_{lk}^{(\ell)} \equiv \check{\hat{\omega}}^{(\ell)}$ or $\hat{\omega}_{lk}^{(\ell)} > \kappa\ddot{\hat{\omega}}^{(\ell)}$.
  2. Call the new merged group $\boldsymbol{\mathcal{C}}_{\min(k,l)}$ and decrease the label index.
  3. Using (9), update the pairwise overlap measures that have changed as a result of the merges. Call the updated measures $\hat{\omega}_{\boldsymbol{\mathcal{C}}_k\boldsymbol{\mathcal{C}}_l}^{(\ell+1)}$.
  4. Obtain the updated overlap matrix (call it $\hat{\boldsymbol{\Omega}}^{(\ell+1)}$) and calculate the updated generalized overlap $\ddot{\hat{\omega}}^{(\ell+1)}$. Set $\ell \leftarrow \ell + 1$.

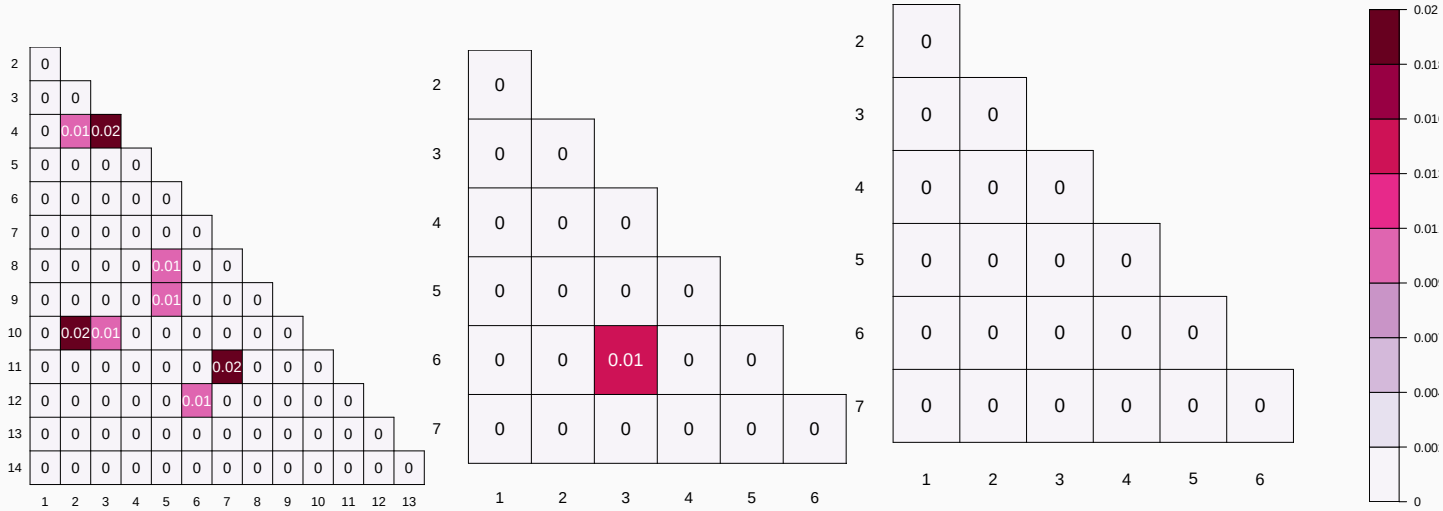
Merging phase (cont'd)

1. The merging phase terminates if either $\ddot{\hat{\omega}}^{(\ell)} > \ddot{\hat{\omega}}^{(\ell-1)}$, or $\ddot{\hat{\omega}}^{(\ell)} \approx 0$, or $\ddot{\hat{\omega}}^{(\ell)} \approx \ddot{\check{\omega}}^{(\ell)}$. The terminating $\hat{K}$ is the estimated $\hat{C}$ of (1).
2. *Final clustering solution:* The grouping $\{\boldsymbol{\mathcal{C}}_1, \boldsymbol{\mathcal{C}}_2, \ldots, \boldsymbol{\mathcal{C}}_{\hat{C}}\}$ at the end of the merging phase is the final partition of the dataset. We therefore have a total of $\hat{C}$ general-shaped groups in the dataset.

Example: `Aggregation` dataset



(a) $\hat{K} = 14$ **(b)** $\hat{C} = 8$ **(c)** $\hat{C} = 7$



(d) $\ddot{\hat{\omega}} = 0.0013$ **(e)** $\ddot{\hat{\omega}} = 0.0008$ **(f)** $\ddot{\hat{\omega}} = 10^{-6}$

Competing methods

Method	Author	Purpose
K-mH	Peterson et al. (2018)	Syncytial
MMC	Baudry et al. (2010)	Syncytial
DEMP	Hennig (2010)	Syncytial
DEMP+	Melnykov (2016)	Syncytial
EAC	Fred and Jain (2005)	Frequency
GSL-NN	Stuetzle and Nugent (2010)	Connectivity
Spectral	-	Connectivity
kernel k-means	-	Connectivity
DBSCAN*	Campello et al. (2013)	Connectivity
Density peaks	Rodriguez and Laio (2014)	Mode
PGMM	McNicholas and Murphy (2008)	Model-based
MSAL	Franczak et al. (2013)	Model-based
MGHD	Browne and McNicholas (2015)	Model-based

Shape datasets used in the two-dimensional evaluations

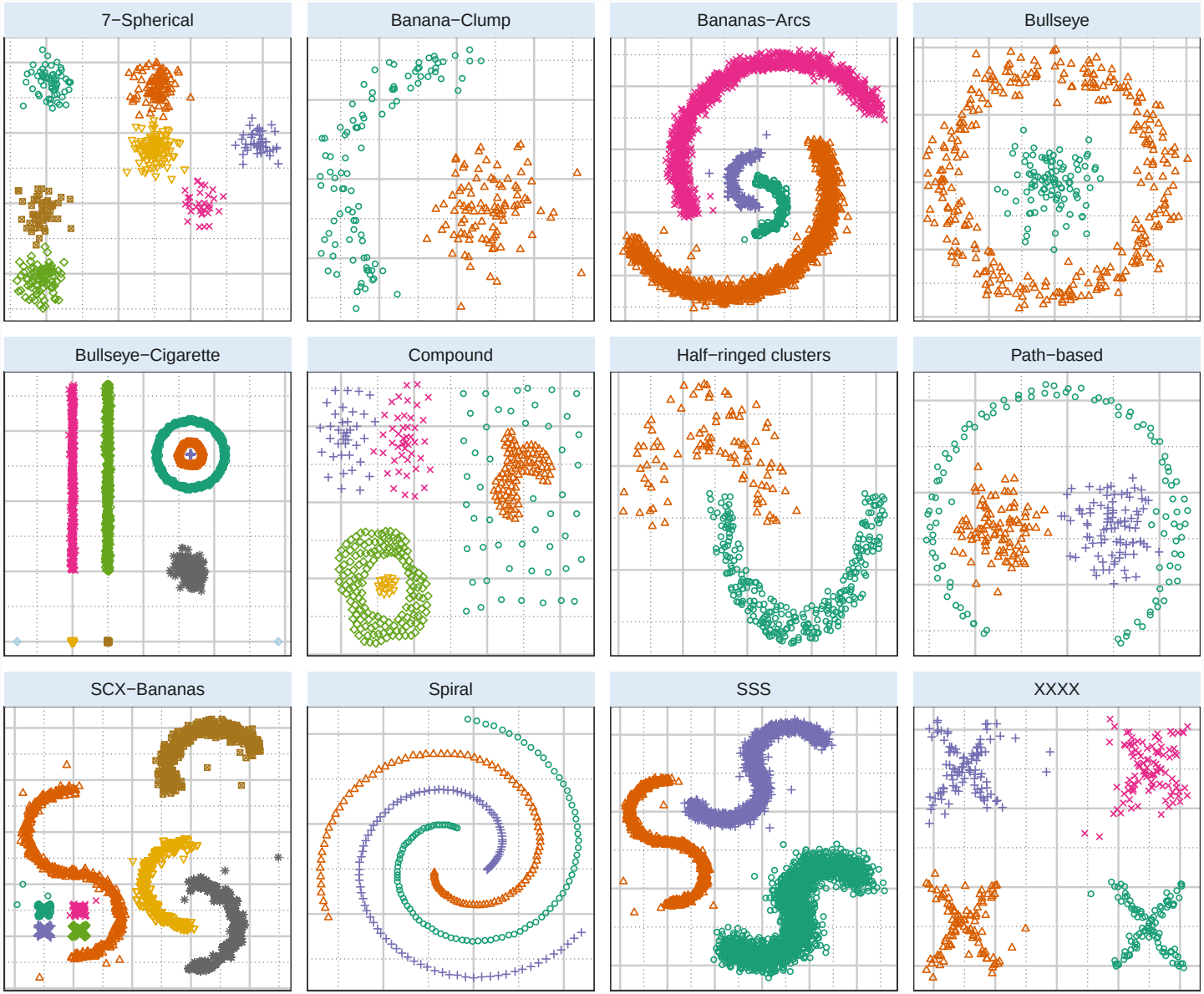
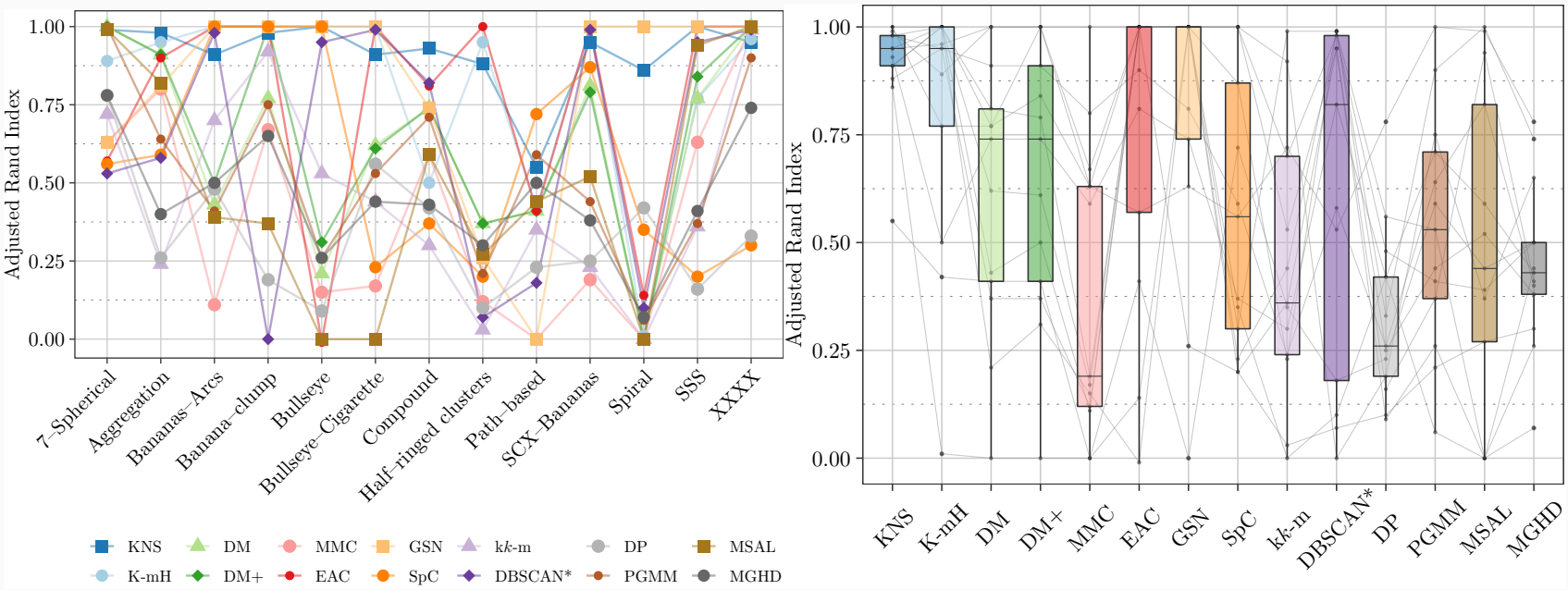


Figure 1: Shape datasets used in the two-dimensional performance

Two-dimensional datasets results



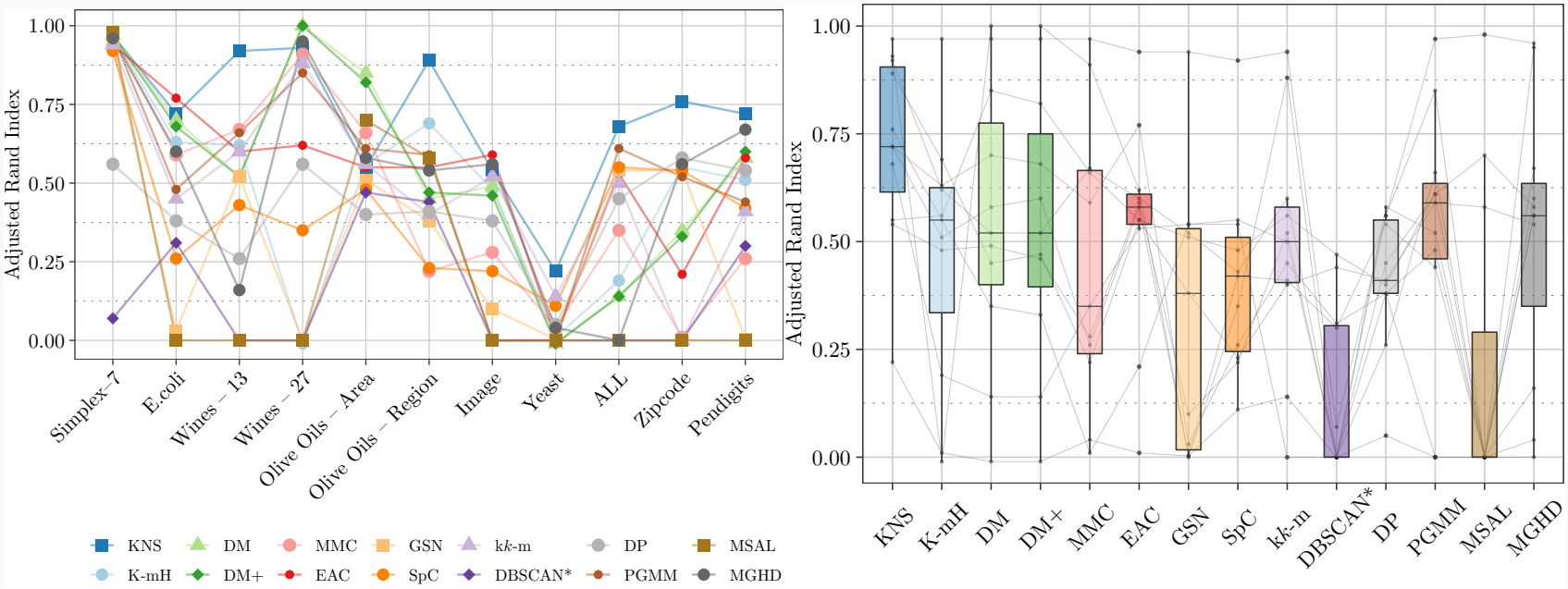
(a) Performance by dataset **(b)** Performance by method

Figure 2: Performance of KNOB-SynC (abbreviation: KNS), K-mH, DEMP (DM), DEMP+ (DM+), MMC, EAC, GSL-NN (GSN), spectral clustering (SpC), kernel k-means (kk-m), DBSCAN*, DP, PGMM, MSAL and MGHD.

High-dimensional datasets

Dataset	(n, p, K, m)
Simplex-7	$(560, 7, 7, 7)$
E.coli	$(336, 7, 7, 5)$
Wines-13	$(178, 13, 3, 17)$
Wines-27	$(178, 27, 3, 26)$
Olive Oils-Area	$(572, 8, 9, 8)$
Olive Oils-Region	$(572, 8, 3, 8)$
Image	$(2310, 19, 7, 8)$
Yeast	$(1484, 8, 10, 6)$
ALL	$(215, 1000, 7, 42)$
Zipcode	$(2000, 256, 10, 33)$
Pendigits	$(10992, 16, 10, 18)$

High-dimensional datasets



(a) Performance by dataset **(b)** Performance by method

Figure 3: Performance of KNOB-SynC (abbreviation: KNS), K-mH, DEMP (DM), DEMP+ (DM+), MMC, EAC, GSL-NN (GSN), spectral clustering (SpC), kernel k-means (kk-m), DBSCAN*, DP, PGMM, MSAL and MGHD.

KNOB-SynC in the presence of scatter

- Maitra and Ramler (2009) developed the k-clips algorithm for k-means clustering in the presence of scatter.
- We use the first 100 images of the Olivetti faces database Samaria and Harter (1994) that were used by Rodriguez and Laio (2014). These are 10 faces each of 10 individuals taken at different angles and under different light conditions.
- Each 112×92 image has a total of 10,304 pixels so we use the first 37 KPCs.
- We started with 70 initial groups. KNOB-SynC's found 9 large groups, 5 small groups and 1 scatter and $\mathcal{R} = 0.902$.

Olivetti faces database

(a) KNOBSynC: $\mathcal{R} = 0.90$

(b) DP: $\mathcal{R} = 0.22$

Figure 4: Clusters of the first 100 images in the Olivetti database obtained by (a) KNOB-SynC and (b) DP.

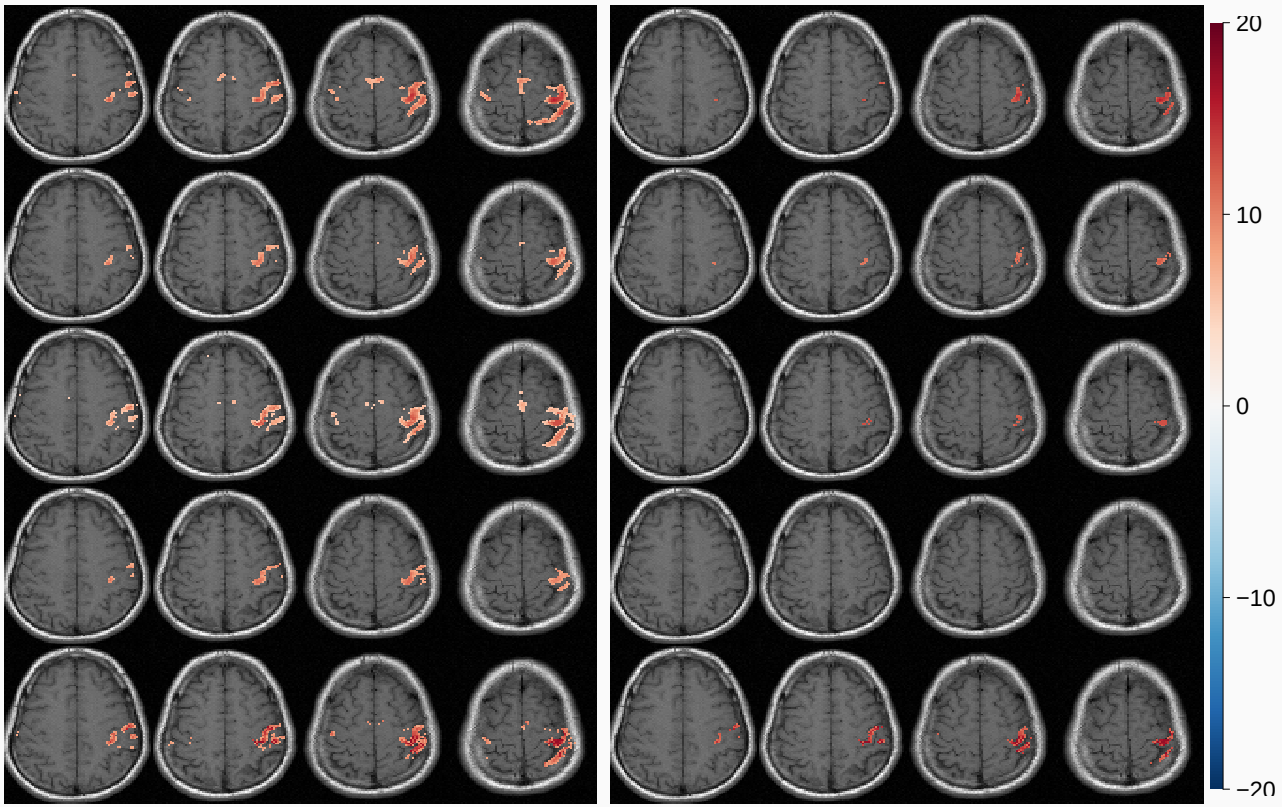
Activation detection in a fMRI finger-tapping task experiment

- One objective of fMRI is to determine cerebral regions that respond to a task or particular stimulus.
- A typical approach relates, the observed Blood Oxygen Level Dependent (BOLD) time course sequence at each image voxel to the expected BOLD response by fitting a general linear model.
- Attempts to use clustering algorithms have been made, like k-means but in general, is not a good performer.

Activation detection in a fMRI finger-tapping task experiment

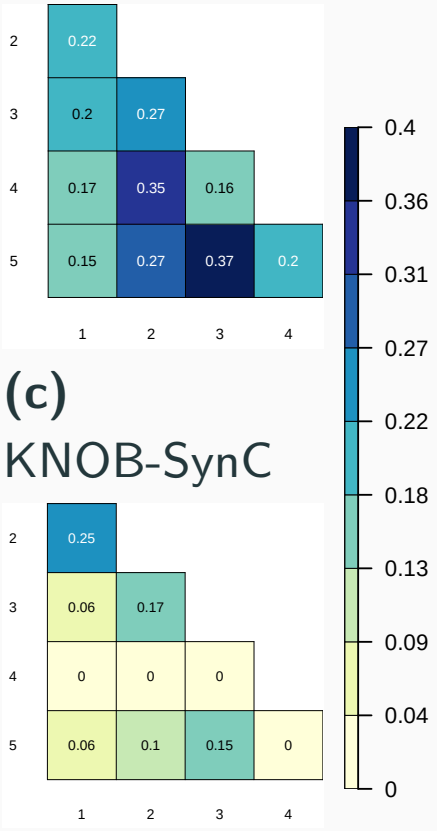
- This experiment is from a right-hand finger-tapping experiment of a right-hand-dominant male and was acquired over twelve regularly-spaced sessions over the course of two months.
- At each of the $n = 179364$ voxels, we computed a Z-scores to test the hypothesis that the expected BOLD levels are significantly related to the right-hand tapping at a voxel.

Activation detection in a fMRI finger-tapping task experiment



(a) KNOB-SynC

(b) AR-FAST



(c) KNOB-SynC

(d) AR-FAST

Conclusion and further work

- This work has proposed a syncytial clustering algorithm called KNOB-SynC that merges groups found by standard clustering algorithms.
- It does so in a fully data-driven and objective way.
- KNOB-SynC can be implemented using the `R` package RSynC available at *https://github.com/ialmodovar/RSynC*.
- Extending KNOB-SynC to a semi-supervised framework.