Robust clustering in (moderately) high dimensional cases

International Conference on Robust Statistics 2025, Stresa, Italy

Luis Angel García-Escudero

Universidad de Valladolid

(joint work with Agustín Mayo-Iscar and Lucía Trapote)

Index

Robust clustering based on trimming

Why robust clustering?

- Outliers are known to be **problematic** in Cluster Analysis:
  - ◇ Relevant and well-defined clusters incorrectly merged
  - ◇ Spurious clusters detected

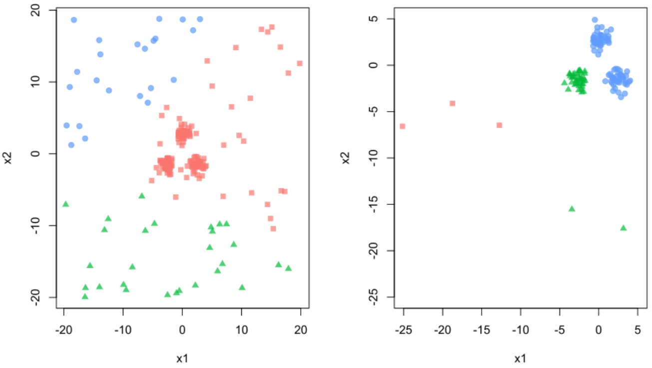


Figure: Impact of contamination on 3-means

Why robust clustering?

- Outliers can be considered as clusters in themselves, suggesting an increase in the number of clusters G
  - ◇ Not always the best strategy (and sometimes infeasible...)

Why robust clustering?

- Outliers can be considered as clusters in themselves, suggesting an increase in the number of clusters G
  - ◇ Not always the best strategy (and sometimes infeasible...)
- **Cluster Analysis/Anomaly Detection** are **related** topics:
  - ◇ The first finds crowds of data points, while the second aims to detect observations far from these crowds
  - ◇ An unified treatment: **Robust Clustering**

Why robust clustering?

- Outliers can be considered as clusters in themselves, suggesting an increase in the number of clusters G
  - ◇ Not always the best strategy (and sometimes infeasible...)
- **Cluster Analysis/Anomaly Detection** are **related** topics:
  - ◇ The first finds crowds of data points, while the second aims to detect observations far from these crowds
  - ◇ An unified treatment: **Robust Clustering**
- Clustered outliers are harmful to (even robust) statistical methods but can be easily detected through clustering

Robust clustering approaches

- Different approaches for robust clustering:

Robust clustering approaches

- Different approaches for robust clustering:
  1. Noise and outliers accommodated by heavy-tailed components: mixtures of t-distributions [Peel and McLachlan, 2000] or mixtures of contaminated Gaussian distributions [Punzo and McNicholas 2016]

Robust clustering approaches

- Different approaches for robust clustering:
  1. Noise and outliers accommodated by heavy-tailed components: mixtures of t-distributions [Peel and McLachlan, 2000] or mixtures of contaminated Gaussian distributions [Punzo and McNicholas 2016]
  2. Modelled by a uniformly distributed noise component [Banfield and Raftery 1993; Coretto and Hennig 2016]

Robust clustering approaches

- Different approaches for robust clustering:
  1. Noise and outliers accommodated by heavy-tailed components: mixtures of t-distributions [Peel and McLachlan, 2000] or mixtures of contaminated Gaussian distributions [Punzo and McNicholas 2016]
  2. Modelled by a uniformly distributed noise component [Banfield and Raftery 1993; Coretto and Hennig 2016]
  3. Trimming approach [Cuesta-Albertos et al 1997, Neykov et al 2007, García-Escudero et al 2008]

Robust clustering approaches

- Different approaches for robust clustering:
  1. Noise and outliers accommodated by heavy-tailed components: mixtures of t-distributions [Peel and McLachlan, 2000] or mixtures of contaminated Gaussian distributions [Punzo and McNicholas 2016]
  2. Modelled by a uniformly distributed noise component [Banfield and Raftery 1993; Coretto and Hennig 2016]
  3. Trimming approach [Cuesta-Albertos et al 1997, Neykov et al 2007, García-Escudero et al 2008]
- We **focus** on that robust clustering approach **based on trimming**

Trimming and robustness

- Standard statistical tools are applied in **trimming** after (hopefully) discarding outliers within the fraction α of trimmed observations $\rightsquigarrow$ Easy interpretation

Trimming and robustness

- Standard statistical tools are applied in **trimming** after (hopefully) discarding outliers within the fraction α of trimmed observations $\rightsquigarrow$ Easy interpretation
- Trimming **self-determined** by data [Rousseeuw 1984, 1985]:
  - ◇ LTS and LMS in regression
  - ◇ MVE and MCD in location-and-scatter estimation

Trimming and robustness

- Standard statistical tools are applied in **trimming** after (hopefully) discarding outliers within the fraction α of trimmed observations $\rightsquigarrow$ Easy interpretation
- Trimming **self-determined** by data [Rousseeuw 1984, 1985]:
  - ◇ LTS and LMS in regression
  - ◇ MVE and MCD in location-and-scatter estimation
- The use of **C-steps (concentration steps)** [Rousseeuw and van Driessen 1999] forms the basis for its practical application

Robust clustering based on trimming

ADVANCED REVIEW

WIREs COMPUTATIONAL STATISTICS WILEY

Robust clustering based on trimming

Luis A. García-Escudero | **Agustín Mayo-Iscar**

Department of Statistics and Operation Research and IMUVA, University of Valladolid, Valladolid, Spain

Correspondence
Luis A. García-Escudero, Department of Statistics and Operation Research and IMUVA, University of Valladolid,

Abstract
Clustering is one of the most widely used unsupervised learning techniques. However, it is well-known that outliers can have a significantly adverse impact on commonly applied clustering methods. On the other hand, clustered outliers can be particularly detrimental to (even robust) statistical procedures. Therefore, it makes sense to combine concepts from Robust Statistics and Clus-

Figure: WIREs Comp Stat 2024

Trimmed K-means [Cuesta-Albertos, Gordaliza and Matrán 1997]

- Given $\{x_1, ..., x_n\} \subset \mathbb{R}^p$, the **trimmed k-means** search for G centers $\mu_1, ..., \mu_G$ and a partition

$$\{R_0, R_1, ..., R_G\} \text{ of } \{1, 2, ..., n\}$$

with

$$\#R_0 = [n\alpha]$$

minimizing

$$\sum_{g=1}^{G} \sum_{i \in R_g} \|x_i - \mu_g\|^2$$

Trimmed K-means [Cuesta-Albertos, Gordaliza and Matrán 1997]

- Given $\{x_1, ..., x_n\} \subset \mathbb{R}^p$, the **trimmed k-means** search for G centers $\mu_1, ..., \mu_G$ and a partition

$$\{R_0, R_1, ..., R_G\} \text{ of } \{1, 2, ..., n\}$$

with

$$\#R_0 = [n\alpha]$$

minimizing

$$\sum_{g=1}^{G} \sum_{i \in R_g} \|x_i - \mu_g\|^2$$

- $R_1, ..., R_G$ gives the partition into G clusters, but a fraction α of observations (those with indexes in R_0) are trimmed

Trimmed K-means

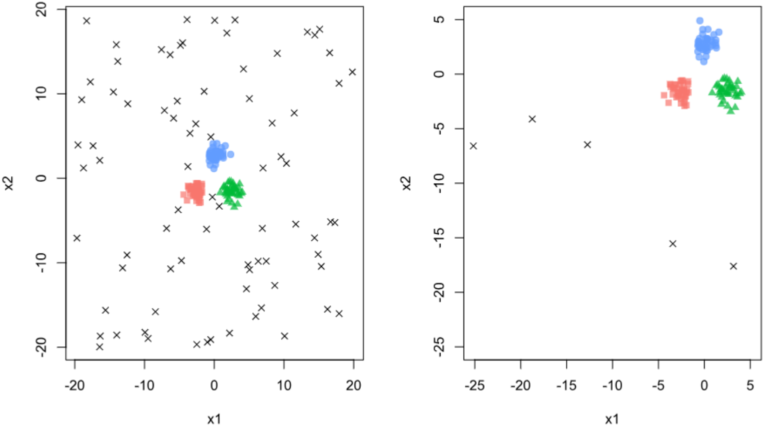


Figure: Trimmed 3-means with $\alpha = 0.05$. Trimmed points as "$\times$"

Trimmed K-means alg. [G-E, Gordaliza and Matrán 2003]

1. *Random initializations*:
   G random observations $\rightsquigarrow$ Initial centers $\mu_1^0, \ldots, \mu_G^0$

Trimmed K-means alg. [G-E, Gordaliza and Matrán 2003]

1. *Random initializations*:
   G random observations $\rightsquigarrow$ Initial centers $\mu_1^0, \dots, \mu_G^0$
2. *C-steps*:
   - **2.1** Take
     $$D_{ig} = \|x_i - \mu_g^{t-1}\|^2 \text{ and } D_i = \min_{g=1,\dots,G} D_{ig}$$
     and $D_{(1)} \leq D_{(2)} \leq \dots \leq D_{(n)}$ to get
     $$R_g = \{i : D_{ig} = D_i \text{ and } D_i \leq D_{([n(1-\alpha)])}\}$$
     for $g = 1, \dots, G$
   - **2.2** Update centers $\mu_g^t = \mathbf{avg}\{x_i : i \in R_g\}$

Trimmed K-means alg. [*G-E*, Gordaliza and Matrán 2003]

1. *Random initializations*:
   G random observations $\rightsquigarrow$ Initial centers $\mu_1^0, \ldots, \mu_G^0$
2. *C-steps*:
   - **2.1** Take
   $$D_{ig} = \|x_i - \mu_g^{t-1}\|^2 \text{ and } D_i = \min_{g=1,\ldots,G} D_{ig}$$
   and $D_{(1)} \leq D_{(2)} \leq \ldots \leq D_{(n)}$ to get
   $$R_g = \{i : D_{ig} = D_i \text{ and } D_i \leq D_{([n(1-\alpha)])}\}$$
   for $g = 1, \ldots, G$
   - **2.2** Update centers $\mu_g^t = \textbf{avg}\{x_i : i \in R_g\}$
3. *Output:* that with the minimum value of the target function.

Trimmed K-means alg. [*G-E*, Gordaliza and Matrán 2003]

1. *Random initializations*:
   G random observations $\rightsquigarrow$ Initial centers $\mu_1^0, \ldots, \mu_G^0$
2. *C-steps*:
   2.1 Take
   $$D_{ig} = \|x_i - \mu_g^{t-1}\|^2 \text{ and } D_i = \min_{g=1,\ldots,G} D_{ig}$$
   and $D_{(1)} \leq D_{(2)} \leq \ldots \leq D_{(n)}$ to get
   $$R_g = \{i : D_{ig} = D_i \text{ and } D_i \leq D_{([n(1-\alpha)])}\}$$
   for $g = 1, \ldots, G$
   2.2 Update centers $\mu_g^t = \textbf{avg}\{x_i : i \in R_g\}$
3. *Output:* that with the minimum value of the target function.

- It reduces to Lloyd's k-means algorithm when $\alpha = 0$

TCLUST as an extension of trimmed k-means

- Trimmed k-means favours spherical/equally scattered clusters

TCLUST as an extension of trimmed k-means

- Trimmed k-means favours spherical/equally scattered clusters
- More **flexible** clustering associated to G **normal** components with location $\{\mu_g\}_{g=1}^G$ and **scatter** matrices $\{\Sigma_g\}_{g=1}^G$

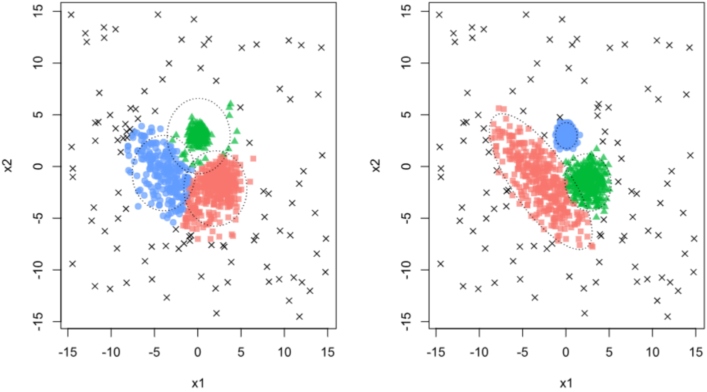


Figure: Trimmed 3-means (left). TCLUST $G = 3$, $c = 12$ and $\alpha = 0.05$ (right)

TCLUST [*G-E*, Gordaliza, Matrán and Mayo-Iscar 2008]

- **TCLUST** searches for centers $\mu_1, \ldots, \mu_G$, **scatter matrices** $\Sigma_1, \ldots, \Sigma_G$, **weights** $\pi_1, \ldots, \pi_g$ (with $\sum_{g=1}^{G} \pi_g = 1$), and a partition $\{R_0, R_1, \ldots, R_G\}$ with $\#R_0 = [n\alpha]$ maximizing

$$\sum_{g=1}^{G} \sum_{i \in R_g} \log \left[\pi_g \phi(x_i; \mu_g, \Sigma_g) \right],$$

where $\phi(\cdot; \mu, \Sigma)$ is the *p.d.f.* of a *p*-variate normal

TCLUST's eigenvalues-ratio constraint

- Another important ingredient of TCLUST is the **eigenvalues-ratio constraint**:

$$\frac{\max_{g=1,\ldots,G,j=1,\ldots,p} \lambda_j(\Sigma_g)}{\min_{g=1,\ldots,G,j=1,\ldots,p} \lambda_j(\Sigma_g)} \leq c,$$

where $\{\lambda_j(\Sigma)\}_{j=1}^p$ is the set of eigenvalues of Σ and $c \geq 1$

TCLUST's eigenvalues-ratio constraint

- Another important ingredient of TCLUST is the **eigenvalues-ratio constraint**:

$$\frac{\max_{g=1,\dots,G,j=1,\dots,p} \lambda_j(\Sigma_g)}{\min_{g=1,\dots,G,j=1,\dots,p} \lambda_j(\Sigma_g)} \leq c,$$

where $\{\lambda_j(\Sigma)\}_{j=1}^p$ is the set of eigenvalues of Σ and $c \geq 1$

- Trimmed K-means when $c = 1$ (and $\pi_1 = \dots = \pi_G$)

TCLUST's eigenvalues-ratio constraint

- Another important ingredient of TCLUST is the **eigenvalues-ratio constraint**:

$$\frac{\max_{g=1,\ldots,G,j=1,\ldots,p} \lambda_j(\Sigma_g)}{\min_{g=1,\ldots,G,j=1,\ldots,p} \lambda_j(\Sigma_g)} \leq c,$$

  where $\{\lambda_j(\Sigma)\}_{j=1}^p$ is the set of eigenvalues of Σ and $c \geq 1$
- Trimmed K-means when $c = 1$ (and $\pi_1 = ... = \pi_G$)
- Any $c < \infty$ makes the constrained maximization of the trimmed likelihood well-defined (target function unbounded when $\mu_g = x_i$ and $|\Sigma_g| \downarrow 0$)

TCLUST's eigenvalues-ratio constraint

- Another important ingredient of TCLUST is the **eigenvalues-ratio constraint**:

$$\frac{\max_{g=1,\ldots,G,j=1,\ldots,p} \lambda_j(\Sigma_g)}{\min_{g=1,\ldots,G,j=1,\ldots,p} \lambda_j(\Sigma_g)} \leq c,$$

  where $\{\lambda_j(\Sigma)\}_{j=1}^{p}$ is the set of eigenvalues of Σ and $c \geq 1$
- Trimmed K-means when $c = 1$ (and $\pi_1 = ... = \pi_G$)
- Any $c < \infty$ makes the constrained maximization of the trimmed likelihood well-defined (target function unbounded when $\mu_g = x_i$ and $|\Sigma_g| \downarrow 0$)
- Also prevents detecting "spurious" clusters

Spurious clusters

- A different, but also relevant, source of **lack of robustness**
- **Spurious clusters** (non-interesting clusters formed by a few almost collinear observations):

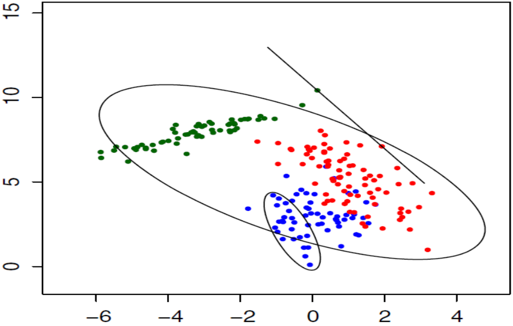


Figure: A spurious cluster detected with $|\Sigma_g| \approx 0$

Spurious clusters

- A different, but also relevant, source of **lack of robustness**
- **Spurious clusters** (non-interesting clusters formed by a few almost collinear observations):

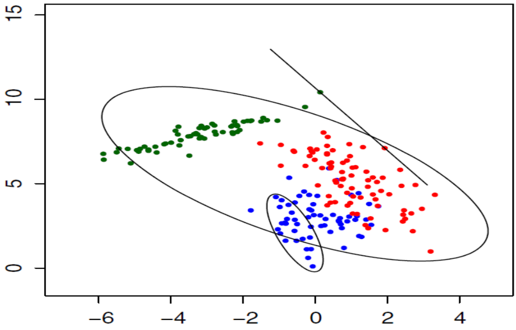


Figure: A spurious cluster detected with $|\Sigma_g| \approx 0$

MCD and trimmed mixture likelihoods

- MCD in the case $G = 1$ (for a large c)

MCD and trimmed mixture likelihoods

- MCD in the case $G = 1$ (for a large c)
- **Trimmed mixture likelihoods** [Neykov, Filzmoser, Dimova and Neytchev 2007; *G-E*, Gordaliza and Mayo-Iscar 2014] maximizing

$$\sum_{i \in R} \log \left[\sum_{g=1}^{G} p_g \phi(x_i; m_g, S_g) \right]$$

with $\#R = [n(1-\alpha)]$

TCLUST algorithm

1. *Random initializations*:
   $G \times (p+1)$ random observations $\rightsquigarrow \mu_1^0, \ldots, \mu_G^0$ and $\Sigma_1^0, \ldots, \Sigma_G^0$

TCLUST algorithm

1. *Random initializations*:
   $G \times (p+1)$ random observations $\rightsquigarrow \mu_1^0, \ldots, \mu_G^0$ and $\Sigma_1^0, \ldots, \Sigma_G^0$
2. *C-steps:*
   2.1 Take

$$D_{ig} = \pi_g^{t-1}\phi(x_i, \mu_g^{t-1}, \Sigma_g^{t-1}) \text{ and } D_i = \max_{g=1,\ldots,G} D_{ig}$$

and $D_{(1)} \leq D_{(2)} \leq \ldots \leq D_{(n)}$ to get

$$R_g = \{i : D_{ig} = D_i \text{ and } D_i \geq D_{([n\alpha])}\}$$

TCLUST algorithm

1. *Random initializations*:
   $G \times (p+1)$ random observations $\rightsquigarrow \mu_1^0, \ldots, \mu_G^0$ and $\Sigma_1^0, \ldots, \Sigma_G^0$
2. *C-steps:*

2.1 Take

$$D_{ig} = \pi_g^{t-1} \phi(x_i, \mu_g^{t-1}, \Sigma_g^{t-1}) \text{ and } D_i = \max_{g=1,\ldots,G} D_{ig}$$

and $D_{(1)} \leq D_{(2)} \leq \ldots \leq D_{(n)}$ to get

$$R_g = \{ i : D_{ig} = D_i \text{ and } D_i \geq D_{([n\alpha])} \}$$

2.2 **MLE**$\{x_i : i \in R_g\} \rightsquigarrow \pi_g^t,\ \mu_g^t$ and S_g

TCLUST algorithm

1. *Random initializations*:
   $G \times (p+1)$ random observations $\rightsquigarrow \mu_1^0, \ldots, \mu_G^0$ and $\Sigma_1^0, \ldots, \Sigma_G^0$
2. *C-steps:*

   2.1 Take

$$D_{ig} = \pi_g^{t-1} \phi(x_i, \mu_g^{t-1}, \Sigma_g^{t-1}) \text{ and } D_i = \max_{g=1,\ldots,G} D_{ig}$$

   and $D_{(1)} \leq D_{(2)} \leq \ldots \leq D_{(n)}$ to get

$$R_g = \{i : D_{ig} = D_i \text{ and } D_i \geq D_{([n\alpha])}\}$$

   2.2 **MLE**$\{x_i : i \in R_g\} \rightsquigarrow \pi_g^t, \mu_g^t$ and S_g

   2.3 Eigenvalue-ratio constraint imposed on the S_g to update Σ_g^t

TCLUST algorithm

1. *Random initializations*:
   $G \times (p+1)$ random observations $\rightsquigarrow \mu_1^0, \ldots, \mu_G^0$ and $\Sigma_1^0, \ldots, \Sigma_G^0$
2. *C-steps:*
   2.1 Take
   $$D_{ig} = \pi_g^{t-1}\phi(x_i, \mu_g^{t-1}, \Sigma_g^{t-1}) \text{ and } D_i = \max_{g=1,\ldots,G} D_{ig}$$
   and $D_{(1)} \leq D_{(2)} \leq \ldots \leq D_{(n)}$ to get
   $$R_g = \{i : D_{ig} = D_i \text{ and } D_i \geq D_{([n\alpha])}\}$$
   2.2 **MLE**$\{x_i : i \in R_g\} \rightsquigarrow \pi_g^t, \mu_g^t$ and S_g
   2.3 Eigenvalue-ratio constraint imposed on the S_g to update Σ_g^t
3. *Output* that with the maximum value

Optimal eigenvalues truncation

- The **optimal truncation operator**:

$$S_g = \mathbf{cov}\{x_i : i \in R_g\} = U_g \mathrm{diag}(d_{g1}, ..., d_{gp}) U_g',$$

with $U_g' U_g = \mathbf{I}_p$.

Optimal eigenvalues truncation

- The **optimal truncation operator**:

$$S_g = \mathbf{cov}\{x_i : i \in R_g\} = U_g \text{diag}(d_{g1}, ..., d_{gp}) U_g',$$

with $U_g' U_g = \mathbf{I}_p$. If

$$[d_{gl}]_m = \max\{m, \min\{d_{gl}, c \cdot m\}\},$$

Optimal eigenvalues truncation

- The **optimal truncation operator**:

$$S_g = \mathbf{cov}\{x_i : i \in R_g\} = U_g \text{diag}(d_{g1}, ..., d_{gp}) U_g',$$

with $U_g' U_g = \mathbf{I}_p$. If

$$[d_{gl}]_m = \max\{m, \min\{d_{gl}, c \cdot m\}\},$$

$n_g = \# R_g$, and

$$m^* = \arg\min_m \sum_{g=1}^{G} n_g \sum_{l=1}^{p} \left(\log[d_{gl}]_m + \frac{d_{gl}}{\log[d_{gl}]_m} \right),$$

then the optimal update is

$$\Sigma_g^t = U_g \text{diag}([d_{g1}]_{m^*}, ..., [d_{gp}]_{m^*}) U_g'$$

TCLUST's good properties

- **Population** version for a theoretical P and its **empirical** version from an i.i.d sample $\{X_1, ..., X_n\}$ with $X_i \sim P$

TCLUST's good properties

- **Population** version for a theoretical P and its **empirical** version from an i.i.d sample $\{X_1, ..., X_n\}$ with $X_i \sim P$
- **Consistency** of empirical toward population solution

TCLUST's good properties

- **Population** version for a theoretical P and its **empirical** version from an i.i.d sample $\{X_1, ..., X_n\}$ with $X_i \sim P$
- **Consistency** of empirical toward population solution
- **Good robustness** performance:
  - Influence function [Ruwet et al 2012]
  - Breakdown point [Ruwet et al 2013]

TCLUST's good properties

- **Population** version for a theoretical P and its **empirical** version from an i.i.d sample $\{X_1, ..., X_n\}$ with $X_i \sim P$
- **Consistency** of empirical toward population solution
- **Good robustness** performance:
  - Influence function [Ruwet et al 2012]
  - Breakdown point [Ruwet et al 2013]
- An efficient algorithm and **packages** are available:
  - `tclust` package in R [Fritz, *G-E* and Mayo-Iscar 2012]
  - `FSDA` Matlab toolbox [Cerioli, Riani, Atkinson and Corbellini 2018]

TCLUST in higher dimensional cases

Higher dimensions and TCLUST...

- **Higher-dimensional** problems are increasingly **common** in current statistical practice

Higher dimensions and TCLUST...

- **Higher-dimensional** problems are increasingly **common** in current statistical practice
- TCLUST works well for small dimensions, but face **challenges** as p increases:

Higher dimensions and TCLUST...

- **Higher-dimensional** problems are increasingly **common** in current statistical practice
- TCLUST works well for small dimensions, but face **challenges** as p increases:
    1. Difficulties in **initialization**

Higher dimensions and TCLUST...

- **Higher-dimensional** problems are increasingly **common** in current statistical practice
- TCLUST works well for small dimensions, but face **challenges** as p increases:
  1. Difficulties in **initialization**
  2. High **number of parameters** involved

Example: Digits data

- **Digits data:** A sample of $n = 1756$ handwritten digits ("3", "5" and "8") from the US postal services (subset of a dataset from UCI). Each digit is a 16×16 gray level image resulting in $p = 16^2 = 256$ dimensional vectors:

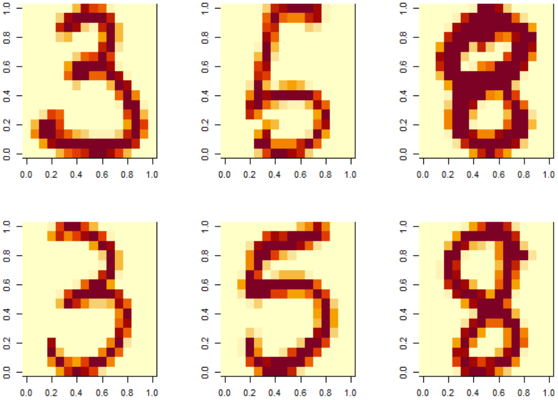


Figure: Six digits images from the Digits dataset

Example: Digits data

- **TCLUST** applied to the **Digits data** with $G = 3$, $c = 12$ and $\alpha = 0.05$ produces (clusters shown in rows, with trimmed images as 0, and actual digits in columns):

	3	5	8
0	21	38	29
1	438	259	81
2	18	61	418
3	181	198	14

- Not satisfactory classification results

Example: Digits data

- **TCLUST** applied to the **Digits data** with $G = 3$, $c = 12$ and $\alpha = 0.05$ produces (clusters shown in rows, with trimmed images as 0, and actual digits in columns):

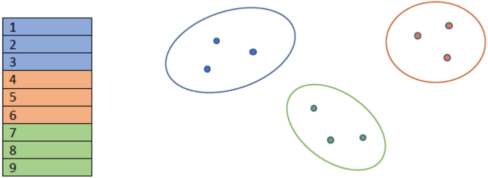
	3	5	8
0	21	38	29
1	438	259	81
2	18	61	418
3	181	198	14

- Not satisfactory classification results
- Requires considerable computing time

Difficulties in initialization

- The algorithm **ideally** requires for initialization a random sub-sample of size $G \times (p+1)$ being outlier-free and properly grouped:

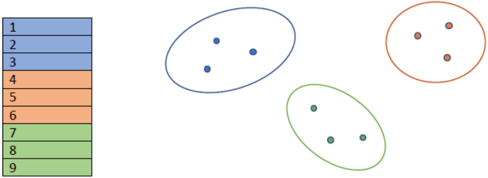
$$G \times (p+1) \text{ observations} \quad \Rightarrow \quad \begin{array}{ll} \text{First } p+1 & \hookrightarrow \text{Cluster } 1 \\ \text{Second } p+1 & \hookrightarrow \text{Cluster } 2 \\ \vdots & \\ G\text{-th } p+1 & \hookrightarrow \text{Cluster } G \end{array}$$



Difficulties in initialization

- The algorithm **ideally** requires for initialization a random sub-sample of size $G \times (p+1)$ being outlier-free and properly grouped:

$$G \times (p+1) \text{ observations} \quad \Rightarrow \quad \begin{array}{ll} \text{First } p+1 & \hookrightarrow \text{Cluster } 1 \\ \text{Second } p+1 & \hookrightarrow \text{Cluster } 2 \\ \vdots & \\ G\text{-th } p+1 & \hookrightarrow \text{Cluster } G \end{array}$$



- That appropriate initializations becomes increasingly **unlikely**

Difficulties in initialization: possible remedies

- **Recent** implementation of the `tclust` package (**version 2.0**: jointly with V. Todorov):
  1. `nstart` random initializations with small `niter1` of C-steps
  2. `nkeep` solutions with the largest values which are iterated until convergence or further `niter2` C-steps

Difficulties in initialization: possible remedies

- **Recent** implementation of the `tclust` package (**version 2.0**: jointly with V. Todorov):
  1. `nstart` random initializations with small `niter1` of C-steps
  2. `nkeep` solutions with the largest values which are iterated until convergence or further `niter2` C-steps

- Combining **partially correct** information from random initializations can lead to effective **ensemble initializations** [Alvarez-Esteban et al 2025+]:
  1. Partitions $\{\mathcal{C}_b\}_{b=1}^{\texttt{nstart}}$ resulting from random initializations
  2. Build an affinity matrix A $(n \times n)$ with:

$$A_{ii'} = \frac{1}{\texttt{nstart}} \#\Big\{ b : x_i \text{ and } x_{i'} \text{ co-clustered (and non-trimmed) in } \mathcal{C}_b \Big\}$$

  3. Exploit the information in matrix A for new initializations

Difficulties in the increasing number of parameters

- A **huge number of parameters**:

$$(G-1)+G\cdot p+G\cdot\frac{(p+1)p}{2}.$$

Difficulties in the increasing number of parameters

- A **huge number of parameters**:

$$(G-1)+G\cdot p+G\cdot \frac{(p+1)p}{2}.$$

  - ◇ What makes TCLUST attractive and flexible for low p leads to failure when p grows...

Difficulties in the increasing number of parameters

- A **huge number of parameters**:

$$(G-1)+G\cdot p+G\cdot\frac{(p+1)p}{2}.$$

  - ◇ What makes TCLUST attractive and flexible for low p leads to failure when p grows...
  - ◇ Even more problematic when n is small relative to p

Difficulties in the increasing number of parameters

- A **huge number of parameters**:

$$(G-1)+G\cdot p+G\cdot \frac{(p+1)p}{2}.$$

  - What makes TCLUST attractive and flexible for low p leads to failure when p grows...
  - Even more problematic when n is small relative to p
- The eigenvalue ratio constraint **"regularizes"** the target function when c is small. However, small c's ($c \approx 1$) enforce **trimmed k-means–type** results

Difficulties in the increasing number of parameters

- A **huge number of parameters**:

$$(G-1)+G\cdot p+G\cdot\frac{(p+1)p}{2}.$$

  - What makes TCLUST attractive and flexible for low p leads to failure when p grows...
  - Even more problematic when n is small relative to p
- The eigenvalue ratio constraint **"regularizes"** the target function when c is small. However, small c's ($c \approx 1$) enforce **trimmed k-means–type** results
- More sophisticated constraints [*G-E*, Mayo-Iscar and Riani 2020, 2022]

Robust Linear Grouping method

Dimension reduction and clustering

- Applying PCA and then clustering (tandem approach) is not the best strategy [Chang 1983]

Dimension reduction and clustering

- Applying PCA and then clustering (tandem approach) is not the best strategy [Chang 1983]
- Perform **clustering** and **dimension reduction** simultaneously

Dimension reduction and clustering

- Applying PCA and then clustering (tandem approach) is not the best strategy [Chang 1983]
- Perform **clustering** and **dimension reduction** simultaneously
- Observations clustered around G **affine subspaces**:
  - Mixtures of PPCA [Tipping and Bishop 1997]
  - Mixtures of Factor Analyzers [Ghahramani and Hinton 1997; McLachlan and Peel 2000]

RLG [*G-E*, Gordaliza, San Martín, van Aelst and Zamar 2009]

- The **Robust Linear Grouping (RLG)** searches for G **affine subspaces** $\mathcal{B}_1, \ldots, \mathcal{B}_G$ with intrinsic dimensions $q_1, \ldots, q_G$ and a partition $\{R_0, R_1, \ldots, R_G\}$ with $\#R_0 = [n\alpha]$ minimizing

$$\sum_{g=1}^{G} \sum_{i \in R_g} \|x_i - \mathrm{Pr}_{\mathcal{B}_g}(x_i)\|^2$$

  - ◇ $\mathrm{Pr}_{\mathcal{B}}(x) \rightsquigarrow$ orthogonal projection of $x \in \mathbb{R}^p$ onto subspace $\mathcal{B}$

RLG [G-E, Gordaliza, San Martín, van Aelst and Zamar 2009]

- The **Robust Linear Grouping (RLG)** searches for G **affine subspaces** $\mathcal{B}_1, \dots, \mathcal{B}_G$ with intrinsic dimensions $q_1, \dots, q_G$ and a partition $\{R_0, R_1, \dots, R_G\}$ with $\#R_0 = [n\alpha]$ minimizing

$$\sum_{g=1}^{G} \sum_{i \in R_g} \|x_i - \mathrm{Pr}_{\mathcal{B}_g}(x_i)\|^2$$

  - ◇ $\mathrm{Pr}_{\mathcal{B}}(x) \rightsquigarrow$ orthogonal projection of $x \in \mathbb{R}^p$ onto subspace $\mathcal{B}$
  - ◇ LTS-PCA when $G = 1$ [Maronna 2005; Croux et al 2017]

RLG [*G-E*, Gordaliza, San Martín, van Aelst and Zamar 2009]

- The **Robust Linear Grouping (RLG)** searches for G **affine subspaces** $\mathcal{B}_1, ..., \mathcal{B}_G$ with intrinsic dimensions $q_1, ..., q_G$ and a partition $\{R_0, R_1, ..., R_G\}$ with $\#R_0 = [n\alpha]$ minimizing

$$\sum_{g=1}^{G} \sum_{i \in R_g} \|x_i - \mathrm{Pr}_{\mathcal{B}_g}(x_i)\|^2$$

  - ◇ $\mathrm{Pr}_{\mathcal{B}}(x) \rightsquigarrow$ orthogonal projection of $x \in \mathbb{R}^p$ onto subspace $\mathcal{B}$
  - ◇ LTS-PCA when $G = 1$ [Maronna 2005; Croux et al 2017]
  - ◇ Trimmed k-means when $q_1 = ... = g_G = 0$

RLG algorithm

1. *Random initializations*:
$\sum_{1=1}^{G}(q_g+1)$ random observ. $\rightsquigarrow$ Affine suspaces $\mathcal{B}_1^0$,..., $\mathcal{B}_G^0$

RLG algorithm

1. *Random initializations*:
   $\sum_{1=1}^{G}(q_g+1)$ random observ. $\rightsquigarrow$ Affine suspaces $\mathcal{B}_1^0,\ldots,\mathcal{B}_G^0$
2. *C-steps*:
   2.1 Take

$$D_{ig}=\|x_i-\mathrm{Pr}_{\mathcal{B}_g^{t-1}}(x_i)\|^2 \text{ and } D_i=\min_{g=1,\ldots,G} D_{ig}$$

and $D_{(1)}\leq D_{(2)}\leq \ldots \leq D_{(n)}$ to get

$$R_g=\{i: D_{ig}=D_i \text{ and } D_i\leq D_{([n(1-\alpha)])}\}$$

   2.2 Update $\mathcal{B}_g^t$ based on $\mathbf{PCA}_{q_g}\{x_i : i\in R_g\}$

RLG algorithm

1. *Random initializations*:
$\sum_{1=1}^{G}(q_g+1)$ random observ. $\rightsquigarrow$ Affine suspaces $\mathcal{B}_1^0,\ldots,\mathcal{B}_G^0$
2. *C-steps*:
   2.1 Take
   $$D_{ig}=\|x_i-\mathrm{Pr}_{\mathcal{B}_g^{t-1}}(x_i)\|^2 \text{ and } D_i=\min_{g=1,\ldots,G} D_{ig}$$
   and $D_{(1)}\leq D_{(2)}\leq \ldots \leq D_{(n)}$ to get
   $$R_g=\{i: D_{ig}=D_i \text{ and } D_i\leq D_{([n(1-\alpha)])}\}$$
   2.2 Update $\mathcal{B}_g^t$ based on $\mathbf{PCA}_{q_g}\{x_i: i\in R_g\}$
3. *Output* that with the minimum value of the target function

RLG algorithm

1. *Random initializations*:
   $\sum_{1=1}^{G}(q_g+1)$ random observ. $\rightsquigarrow$ Affine suspaces $\mathcal{B}_1^0,\ldots,\mathcal{B}_G^0$
2. *C-steps*:
   2.1 Take

   $$D_{ig}=\|x_i-\mathrm{Pr}_{\mathcal{B}_g^{t-1}}(x_i)\|^2 \text{ and } D_i=\min_{g=1,\ldots,G} D_{ig}$$

   and $D_{(1)}\leq D_{(2)}\leq \ldots \leq D_{(n)}$ to get

   $$R_g=\{i: D_{ig}=D_i \text{ and } D_i\leq D_{([n(1-\alpha)])}\}$$

   2.2 Update $\mathcal{B}_g^t$ based on $\mathbf{PCA}_{q_g}\{x_i: i\in R_g\}$
3. *Output* that with the minimum value of the target function

- It can be applied using `rlg()` function in `tclust` package

RLG focus on orthogonal errors

- **Isotropic** orthogonal errors and troubles with **intersecting subspaces**:

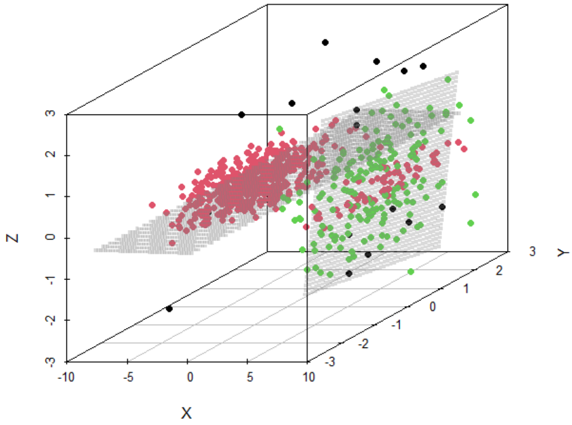


Figure: Troubles with intersecting subspaces ($q_1 = q_2 = 2$)

Trimmed HDDC

Trimmed HDDC method

- A **compromise between TCLUST and RLG**

Trimmed HDDC method

- A **compromise between TCLUST and RLG**
- It arises from the **HDDC** (High Dimensional Data Clustering) approach [Bouveyron, Girard and Schmid 2007; Bouveyron and Brunet-Saumard 2014], where

$$\Sigma_g = U_g \Delta_g U_g',$$

  - U_g is the orthonormal matrix with the eigenvectors of Σ_g
  - Δ_g is a diagonal matrix with the sorted eigenvalues,

but with a special **parsimonious** structure on the Δ_g matrix.

Trimmed HDDC method

- Δ_g has the form

$$\Delta_g = \left(\begin{array}{ccc|ccc} \lambda_{g1} & & 0 & & & \\ & \ddots & & & \mathbf{0} & \\ 0 & & \lambda_{gq_g} & & & \\ \hline & & & \lambda_g & & 0 \\ & \mathbf{0} & & & \ddots & \\ & & & 0 & & \lambda_g \end{array} \right) \begin{array}{l} \left. \vphantom{\begin{array}{c} a \\ b \\ c \end{array}} \right\} q_g \\ \left. \vphantom{\begin{array}{c} a \\ b \\ c \end{array}} \right\} (p - q_g) \end{array}$$

where:

- $\lambda_{gl} \geq \lambda_g$ for $l = 1, ..., q_g$ and $g = 1, ..., G$
- $q_g < p$ for $g = 1, ..., G$

Trimmed HDDC method

- Additional **eigenvalues-ratio constraints** imposed:

$$\frac{\max_{g=1,\ldots,G,j=1,\ldots,q_g} \lambda_{gj}}{\min_{g=1,\ldots,G,j=1,\ldots,q_g} \lambda_{gj}} \leq c_1,$$

and

$$\frac{\max_{g=1,\ldots,G} \lambda_g}{\min_{g=1,\ldots,G} \lambda_g} \leq c_2$$

- The second constraint is the most relevant one

Trimmed HDDC method

- Let $\mathcal{B}_g$ be the affine subspace passing through μ_g and spanned by the first q columns of U_g denoted as $u_{g1}, ..., u_{gq_g}$
- We have

$$\log[\pi_g \phi(x_i; m_g, \Sigma_g)] = \log(\pi_g) - \frac{1}{2}\Bigg(\|\text{Pr}_{\mathcal{B}_g}(x_i) - \mu_g\|^2_{\mathcal{B}_g} + \frac{1}{\lambda_g}\|x_i - \text{Pr}_{\mathcal{B}_g}(x_i)\|^2 + \sum_{l=1}^{q_g} \log(\lambda_{gl}) + (p - q_g)\log(\lambda_g) + p\log(2\pi)\Bigg)$$

Trimmed HDDC method

- Let $\mathcal{B}_g$ be the affine subspace passing through μ_g and spanned by the first q columns of U_g denoted as $u_{g1}, \dots, u_{gq_g}$
- We have

$$\log[\pi_g \phi(x_i; m_g, S_g)] = \log(\pi_g) - \frac{1}{2}\Bigg(\sum_{l=1}^{q_g} \frac{\langle x_i - \mu_g, u_{gl}\rangle^2}{\lambda_{gl}} + \frac{1}{\lambda_g}\Big\| x_i - \mu_g - \sum_{l=1}^{q_g} \langle x_i - \mu_g, u_{gl}\rangle u_{gl} \Big\|^2 + \sum_{l=1}^{q_g} \log(\lambda_{gl}) + (p - q_g)\log(\lambda_g) + p\log(2\pi) \Bigg) \rightsquigarrow \text{Only first } q_g \text{ eigenv.!!!}$$

Trimmed HDDC method

- Let $\mathcal{B}_g$ be the affine subspace passing through μ_g and spanned by the first q columns of U_g denoted as $u_{g1}, \ldots, u_{gq_g}$
- We have

$$\log[\pi_g \phi(x_i; m_g, S_g)] = \log(\pi_g) - \frac{1}{2}\Bigg(\underbrace{\|\mathrm{Pr}_{\mathcal{B}_g}(x_i) - \mu_g\|^2_{\mathcal{B}_g}}_{\text{TCLUST type in } \mathcal{B}_g} + \frac{1}{\lambda_g} \underbrace{\|x_i - \mathrm{Pr}_{\mathcal{B}_g}(x_i)\|^2}_{\text{RLG type in } \mathcal{B}_g^{\perp}} + \sum_{l=1}^{q_g} \log(\lambda_{gl}) + (p - q_g)\log(\lambda_g) + p\log(2\pi)\Bigg)$$

Trimmed HDDC method

- **Summary:**

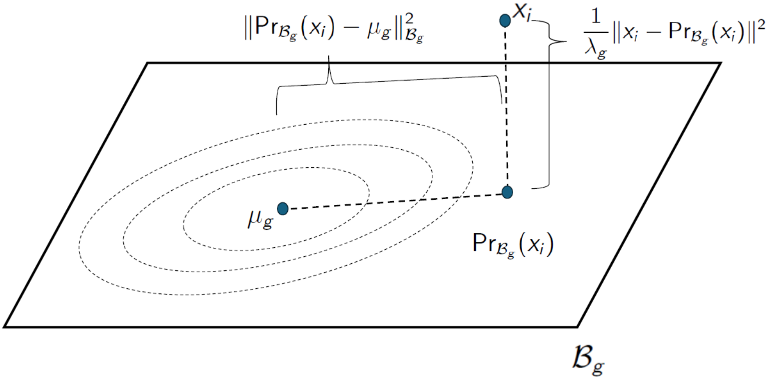


Figure: Trimming x_i with large orthogonal residuals or large distances after projection onto the approximating affine subspace

Trimmed HDDC method

- Trimmed **classification/mixture** likelihoods as

$$\sum_{g=1}^{G} \sum_{i \in R_g} \log \left[\pi_g \phi(x_i; \mu_g, \Sigma_g) \right]$$

and

$$\sum_{i \in R} \log \left[\sum_{q=1}^{G} \pi_g \phi(x_i; \mu_g, \Sigma_g) \right]$$

that (in both cases) require (in the M-step) the maximization of **completed likelihoods**:

$$\sum_{i=1}^{n} \sum_{g=1}^{G} z_{ig} \sum_{i \in R_g} \log \left[\pi_g \phi(x_i; \mu_g, \Sigma_g) \right]$$

Trimmed HDDC algorithm

1. *Random initializations:* *To be explained later...*

Trimmed HDDC algorithm

1. *Random initializations:* *To be explained later...*
2. *C-steps:*
   - 2.1 *Update z_{ig}'s:* Let

$$D_{ig} = \pi_g^{t-1}\phi\Big(x_i, \mu_g^{t-1}, \overbrace{\{u_{gl}^{t-1}, \lambda_{gl}^{t-1}, \lambda_g^{t-1}\}_{l=1}^{q_g}}^{\text{Relevant info from } \Sigma_g^{t-1}}\Big),$$

$D_i = \max_{g=1,\dots,G} D_{ig}$ (clasiff.), $D_i = \sum_{g=1}^{G} D_{ig}$ (mixt.) and

$$z_{ig} = \begin{cases} 1 & \text{if } D_{ig} = D_i \\ 0 & \text{otherwise} \end{cases} \text{ (clasiff.) or } z_{ig} = \frac{D_{ig}}{\sum_{g=1}^{G} D_{ig}} \text{ (mixt.)},$$

but $z_{ig} = 0$, for every $g = 1, ..., G$, if $D_i \leq D_{([n\alpha])}$

Trimmed HDDC algorithm

2.2 *Update parameters:*

2.2.1 Update weights $\pi_g^t = \dfrac{\sum_{g=1}^G z_{ig}}{[n(1-\alpha)]}$

2.2.2 Update means $\mu_g^t = \dfrac{\sum_{g=1}^G z_{ig} x_i}{\sum_{g=1}^G z_{ig}}$

2.2.3 Compute $S_g = \dfrac{1}{\sum_{g=1}^G z_{ig}} \sum_{g=1}^G z_{ig}(x_i - \mu_g^l)(x_i - \mu_g^t)'$ and obtain its eigenvectors u_{g1}^t, ..., $u_{gq_g}^t$ associated to the q_g-th largest eigenvalues $d_{g1}, \dots, d_{gq_g}$ and obtain

$$d_g = \frac{1}{p - q_g}\Big(\text{trace}(S_g) - \sum_{l=1}^{q_g} d_{gl}\Big).$$

Trimmed HDDC algorithm

2.2.4 Impose constraints on

$$\{\{d_{gl}\}_{l=1,\ldots,q_g}\}_{g=1}^{G} \text{ and } \{d_g\}_{g=1}^{G}$$

by applying twice the optimal truncation operation described for TCLUST with constants c_1 and c_2 and return

$$\{\{\lambda_{gl}^{t}\}_{l=1,\ldots,q_g}\}_{g=1}^{G} \text{ and } \{\lambda_g^{t}\}_{g=1}^{G}$$

3. *Output* that with the maximum value of the target function

Trimmed HDDC algorithm

2.2.4 Impose constraints on

$$\{\{d_{gl}\}_{l=1,\dots,q_g}\}_{g=1}^G \text{ and } \{d_g\}_{g=1}^G$$

by applying twice the optimal truncation operation described for TCLUST with constants c_1 and c_2 and return

$$\{\{\lambda_{gl}^t\}_{l=1,\dots,q_g}\}_{g=1}^G \text{ and } \{\lambda_g^t\}_{g=1}^G$$

3. *Output* that with the maximum value of the target function

- Only eigenvalues/eigenvectors associated to the q_g largest eigenvalues required (Arnoldi's method...)

Trimmed HDDC algorithm

2.2.4 Impose constraints on

$$\{\{d_{gl}\}_{l=1,\ldots,q_g}\}_{g=1}^{G} \text{ and } \{d_g\}_{g=1}^{G}$$

by applying twice the optimal truncation operation described for TCLUST with constants c_1 and c_2 and return

$$\{\{\lambda_{gl}^{t}\}_{l=1,\ldots,q_g}\}_{g=1}^{G} \text{ and } \{\lambda_g^{t}\}_{g=1}^{G}$$

3. *Output* that with the maximum value of the target function

- Only eigenvalues/eigenvectors associated to the q_g largest eigenvalues required (Arnoldi's method...)
- TCLUST as a particular case if $q_g = p - 1$ and $c_1 = c_2$ and connections with the trimmed MFA [*G-E* et al 2016]

Initialization step

- Select a random subsample of size $\sum_{g=1}^{G}(q_g+2)$ ($\ll G\times(p+1)$)

Initialization step

- Select a random subsample of size $\sum_{g=1}^{G}(q_g+2)$ ($\ll G\times(p+1)$)
- The minimal set of observations to initialize all parameters...

Initialization step

- Select a random subsample of size $\sum_{g=1}^{G}(q_g+2)$ ($\ll G\times(p+1)$)
- The minimal set of observations to initialize all parameters...
- From observations $x_{i_1},\dots,x_{i_{q_g+2}}$ (in general position):
    - Take $\mu_g^0=\textbf{avg}\{x_{i_1},\dots,x_{i_{q_g+2}}\}$
    - Take $u_{g1}^0,\dots,u_{gq_g}^0$ and $d_{g1},\dots,d_{gq_g}$ associated to the q_g largest eigenvalues of $\textbf{cov}\{x_{i_1},\dots,x_{i_{q_g+2}}\}$
    - d_g equal to the smallest eigenvalue divided by $p-q_g$
    - If $\mathfrak{X}$ is the $(q_g+2)\times p$ matrix whose columns are these observations centred with μ_g^0 then, instead of using $p\times p$ matrix $\mathfrak{X}'\mathfrak{X}$, use the $(q_g+2)\times(q_g+2)$ matrix $\mathfrak{X}\mathfrak{X}'$...

Digits data (revisited)

- **tHDDC** with $q_1 = q_2 = q_3 = 8$, $c_1 = 5$, $c_2 = 2$ and $\alpha = 0.05$ for the **Digits data**:

	0	1	2	3
3	20	29	23	586
5	23	3	510	20
8	44	484	7	7

Digits data (revisited)

- **tHDDC** with $q_1 = q_2 = q_3 = 8$, $c_1 = 5$, $c_2 = 2$ and $\alpha = 0.05$ for the **Digits data**:

	0	1	2	3
3	20	29	23	586
5	23	3	510	20
8	44	484	7	7

- Better performance than TCLUST and RLG

Digits data: outliers

- If $D_{ig} = \pi_g \phi(x_i; \mu_g, \Sigma_g)$ and $D_i = \max_{g=1,\ldots,G} D_{ig}$, the trimmed observations are those with the smallest D_i:

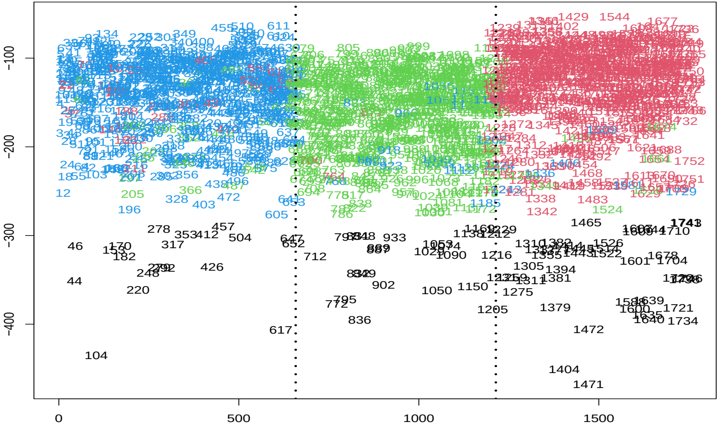


Figure: D_i for $i = 1, \ldots, 1756$ with cluster assignments in colors and black for trimmed

Digits data: trimmed digits

- Some of the observations with the smallest D_i values (trimmed ones):

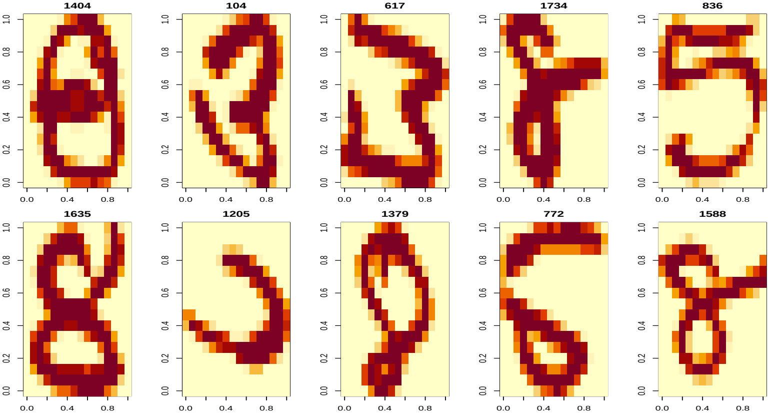


Figure: Some trimmed units

Digits data: estimated centers

- The estimated location vectors μ_1, μ_2 and μ_3:

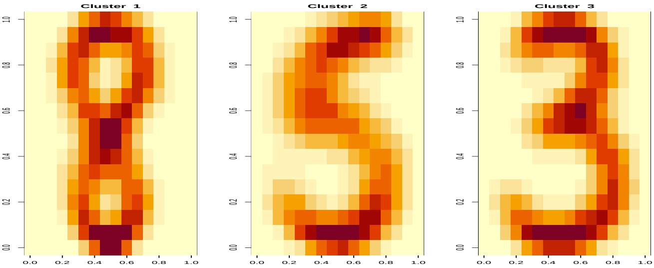


Figure: Estimated clusters' location vectors

Digits data: Loadings

- u_{g1}, u_{g2}, ..., u_{gq_g} loadings interpreted as “variations modes”:

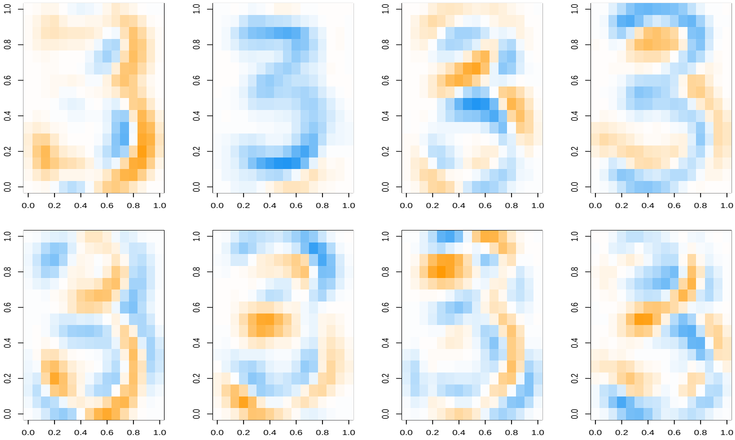


Figure: Loadings ($q_3 = 8$) for the cluster including the 3’s: ■ for positive weights and ■ for negative ones

Digits data: scores

- **Scores in clusters**, $t_{ig,l} = \langle x_i - \mu_g, u_{gl} \rangle$ for $l = 1, \ldots, q_g$, reflect proximity of observations in clusters and also interpretability via the loading vectors values:

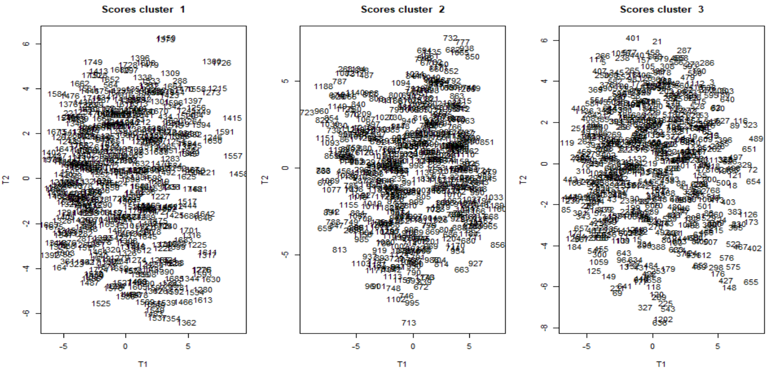


Figure: Scatter plots of the scores ($l = 1, 2$) for the three clusters

Digits data: diagnostic plots

- **Score distances** as

$$\mathrm{SD}_{ig} = \|\mathrm{Pr}_{\mathcal{B}_g}(x_i) - \mu_g\|_{\mathcal{B}_g} = \sqrt{\sum_{l=1}^{q_g} \frac{t_{ig,l}^2}{\lambda_{gl}}}$$

Digits data: diagnostic plots

- **Score distances** as

$$\text{SD}_{ig} = \|\text{Pr}_{\mathcal{B}_g}(x_i) - \mu_g\|_{\mathcal{B}_g} = \sqrt{\sum_{l=1}^{q_g} \frac{t_{ig,l}^2}{\lambda_{gl}}}$$

- **Orthogonal distances** as

$$\text{OD}_{ig} = \frac{1}{\sqrt{\lambda_g}}\|x_i - \text{Pr}_{\mathcal{B}_g}(x_i)\| = \frac{1}{\sqrt{\lambda_g}}\left\|x_i - \mu_g - \sum_{l=1}^{q_g} t_{ig,l} u_{gl}\right\|$$

Digits data: diagnostic plots

- **Score distances** as

$$\mathrm{SD}_{ig} = \|\mathrm{Pr}_{\mathcal{B}_g}(x_i) - \mu_g\|_{\mathcal{B}_g} = \sqrt{\sum_{l=1}^{q_g} \frac{t_{ig,l}^2}{\lambda_{gl}}}$$

- **Orthogonal distances** as

$$\mathrm{OD}_{ig} = \frac{1}{\sqrt{\lambda_g}} \|x_i - \mathrm{Pr}_{\mathcal{B}_g}(x_i)\| = \frac{1}{\sqrt{\lambda_g}} \left\| x_i - \mu_g - \sum_{l=1}^{q_g} t_{ig,l} u_{gl} \right\|$$

- Computed for $A_g = \{i : D_{ig} = D_i\}$ (trimmed observations also included) and **cutoffs** $\sqrt{\chi^2_{q_g;0.975}}$ for the SD_{ig} and $(\widehat{\mu}_{\mathrm{OD}} + \widehat{\sigma}_{\mathrm{OD}} z_{0.975})^{3/2}$ where $\widehat{\mu}_{\mathrm{OD}}$ and $\widehat{\sigma}^2_{\mathrm{OD}}$ are the robust mean and variance based on MCD on $\{\mathrm{OD}_{ig}^{2/3}\}$ for the OD_{ig} [Hubert at al 2005]

Digits data: diagnostic plot example

- Score and orthogonal distances:

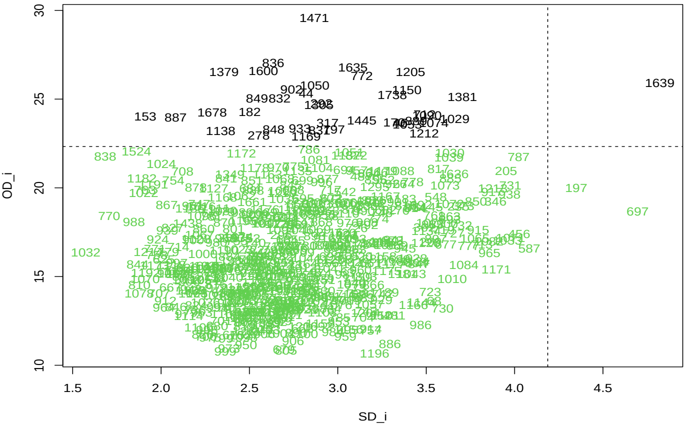


Figure: SD_{ig} and OD_{ig} for g corresponding to the cluster with the 3's

Digits data: diagnostic plot example

- Some of the atypical digits detected in the previous plot:

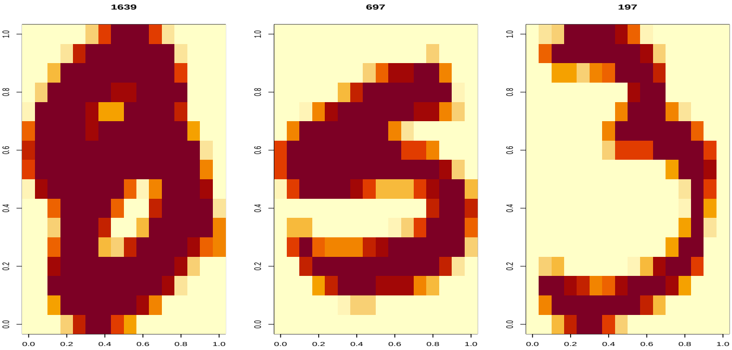


Figure: Some highlighted outliers using SD_{ig} and OD_{ig} for the cluster with the 3's

Digits data: discriminant factors

- From the $D_{ig} = \pi_g \phi(x_i; \mu_g, \Sigma_g)$, let

$$D_{i(1)} \leq D_{i(2)} \leq \ldots \leq D_{i(G)}$$

and define **Discriminant Factors** as

$$\text{DF}(i) = \log \frac{D_{i(G-1)}}{D_{i(G)}} \text{ if } x_i \text{ is not trimmed}$$

Digits data: discriminant factors

- From the $D_{ig} = \pi_g \phi(x_i; \mu_g, \Sigma_g)$, let

$$D_{i(1)} \leq D_{i(2)} \leq \dots \leq D_{i(G)}$$

and define **Discriminant Factors** as

$$\mathrm{DF}(i) = \log \frac{D_{i(G-1)}}{D_{i(G)}} \text{ if } x_i \text{ is not trimmed}$$

- If $D^* = D_{([n\alpha])}$ is the cutoff to label outliers then

$$\mathrm{DF}(i) = \log \frac{D_i}{D^*} \text{ if } x_i \text{ trimmed}$$

Digits data: discriminant factors

- From the $D_{ig} = \pi_g \phi(x_i; \mu_g, \Sigma_g)$, let

$$D_{i(1)} \leq D_{i(2)} \leq ... \leq D_{i(G)}$$

and define **Discriminant Factors** as

$$\mathrm{DF}(i) = \log \frac{D_{i(G-1)}}{D_{i(G)}} \text{ if } x_i \text{ is not trimmed}$$

- If $D^* = D_{([n\alpha])}$ is the cutoff to label outliers then

$$\mathrm{DF}(i) = \log \frac{D_i}{D^*} \text{ if } x_i \text{ trimmed}$$

- $\mathrm{DF}(i) \leq 0$ but $\mathrm{DF}(i) \simeq 0$ for the most **doubtful** assignment decisions.

Digits data: silhouette plot

- **Silhouette plot:**

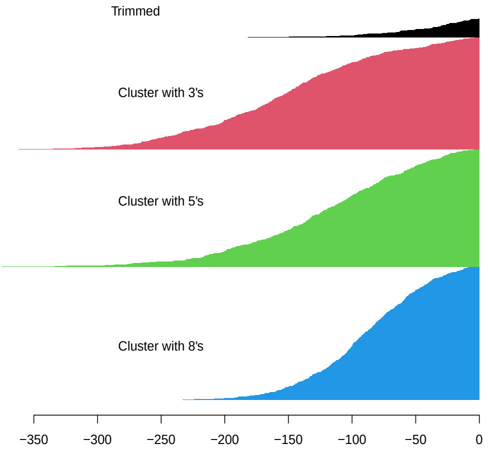


Figure: Silhouette plot with the DF(i) values

Choice of hyperparameters

- q_1,...,q_G (even G...), α, c_1 and c_2???

Choice of hyperparameters

- q_1,...,q_G (even G...), α, c_1 and c_2???
- A **complex** problem... as in others (robust) clustering methods

Choice of hyperparameters

- $q_1,\ldots,q_G$ (even $G\ldots$), α, c_1 and c_2???
- A **complex** problem... as in others (robust) clustering methods
- Cannot be fully automated because user decisions are often required

Choice of hyperparameters

- q_1,...,q_G (even G...), α, c_1 and c_2???
- A **complex** problem... as in others (robust) clustering methods
- Cannot be fully automated because user decisions are often required

- The intrinsic dimensions q_1,...,q_G can be treated as **inner parameters to be estimated** within the iterative steps of the algorithm

Catell procedure estimating q_g's

- **Catell procedure** based on differences $\lambda_{g,l+1} - \lambda_{g,l}$ of the sorted eigenvalues of S_g (with an upper bound $q_{\max}$):

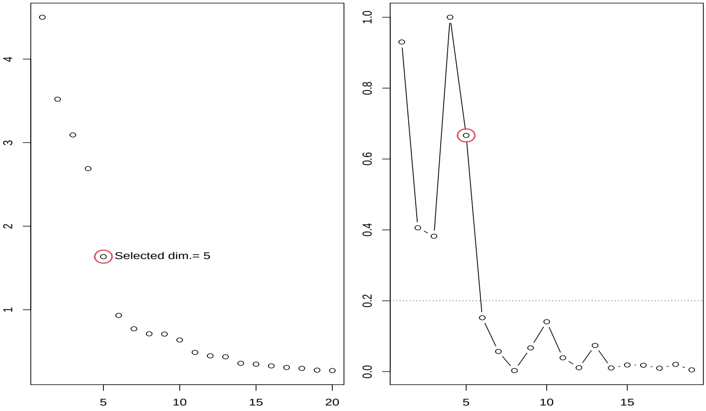


Figure: Estimate q_g as largest index where (normalized) differences exceed a threshold `tresh`: $q_g = 5$ selected with `tresh=0.2`

Catell procedure estimating q_g's

- Estimating the q_g's requires a **BIC-type** [Cerioli et al 2018] modified **target function**:

Catell procedure estimating q_g's

- Estimating the q_g's requires a **BIC-type** [Cerioli et al 2018] modified **target function**:

$$-2 \times \text{trimmed log-likelihood}(q_1, ..., q_G) + \textbf{complexity penalty}$$

Catell procedure estimating q_g's

- Estimating the q_g's requires a **BIC-type** [Cerioli et al 2018] modified **target function**:

$-2\times$ trimmed log-likelihood$(q_1,\ldots,q_G)$ + **complexity penalty**

with **complexity penalty** equal to

$$\log\left([n(1-\alpha)]\right)\Bigg[\underbrace{G-1}_{\text{weights}}+\underbrace{Gp}_{\text{means}}+\underbrace{1+\left(\sum_{g=1}^{G}q_g-1\right)\left(1-\frac{1}{c_1}\right)}_{\text{the largest eigenvalues}}$$

$$\underbrace{+1+(G-1)\left(1-\frac{1}{c_2}\right)}_{\text{the smallest ones}}+\underbrace{\sum_{g=1}^{G}\left(q_g p-\frac{q_g(q_g-1)}{2}\right)}_{\text{orthonormal eigenvec.}}\Bigg]$$

Example I: estimating q_g's

- **Simulated data** set with $n = 1000$ and $p = 200$ where
  - $\mu_1 = 0.2 \cdot \mathbf{1}_{200}$ and $\Sigma_1 = \text{diag}(5, 4, 3, 2, 1, 0.1, ..., 0.1)$: 57% rows
  - $\mu_2 = -0.2 \cdot \mathbf{1}_{200}$ and $\Sigma_2 = \text{diag}(4, 3, 2, 0.15, ..., 0.15)$: 38% rows
  - $x_{ij} \sim \mathcal{U}[-2, 2]$: 5% rows

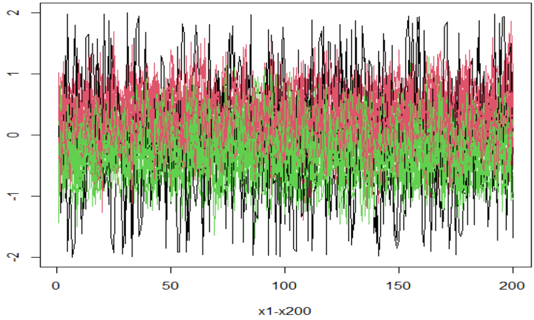


Figure: Line plots of the simulated dataset with $q_1 = 5$ and $q_2 = 3$

Example I: estimating q_g's

- **tHDDC** with $G = 2$, $q_{\max} = 20$, $c_1 = 10$, $c_2 = 2$ and $\alpha = 0.05$ returns $\widehat{q}_1 = 5$ and $\widehat{q}_2 = 3$:

λ_{1l}:	4.7656	4.1874	3.0381	1.9648	1.0376	λ_1 : 0.0983
λ_{2l}:	3.6971	2.9656	2.0139			λ_2 : 0.1482

	1	2	0
0	0	0	50
1	570	0	0
2	0	380	0

Example I: estimating q_g's

- **tHDDC** with $G = 2$, $q_{\max} = 20$, $c_1 = 10$, $c_2 = 2$ and $\alpha = 0.05$ returns $\widehat{q}_1 = 5$ and $\widehat{q}_2 = 3$:

λ_{1l}:	4.7656	4.1874	3.0381	1.9648	1.0376	$\lambda_1 : 0.0983$
λ_{2l}:	3.6971	2.9656	2.0139			$\lambda_2 : 0.1482$

	1	2	0
0	0	0	50
1	570	0	0
2	0	380	0

- **TCLUST** with $G = 2$, $c = 12$ and $\alpha = 0.05$:

	1	2	0	
0	0	2	48	
1	502	90	0	# ×10 computing time
2	68	288	2	

Example II: Digits data

- tHDDC **estimating q_g's** with Catell for the **Digits data** returns $q_1 = q_2 = 10$ and $q_3 = 9$ (with $q_{\max} = 20$ and `tresh=0.05`):

	0	1	2	3
3	25	16	24	593
5	22	8	522	4
8	40	491	2	9

Example II: Digits data

- tHDDC **estimating q_g's** with Catell for the **Digits data** returns $q_1 = q_2 = 10$ and $q_3 = 9$ (with $q_{\max} = 20$ and `tresh=0.05`):

	0	1	2	3
3	25	16	24	593
5	22	8	522	4
8	40	491	2	9

- Even better misclassification rate...

Other issues

Cellwise trimming

- Robustness against **cellwise contamination** [Alqallaf et al 2009]
  - **Cases** $\rightarrow x_i = (x_{i1}, ..., x_{ip})' \in \mathbb{R}^p$ and **Cells** $\rightarrow x_{ij} \in \mathbb{R}$

Cellwise trimming

- Robustness against **cellwise contamination** [Alqallaf et al 2009]
  - ◇ **Cases** $\rightarrow x_i = (x_{i1}, \dots, x_{ip})' \in \mathbb{R}^p$ and **Cells** $\rightarrow x_{ij} \in \mathbb{R}$
- A lot of useful information sacrificed by casewise trimming:

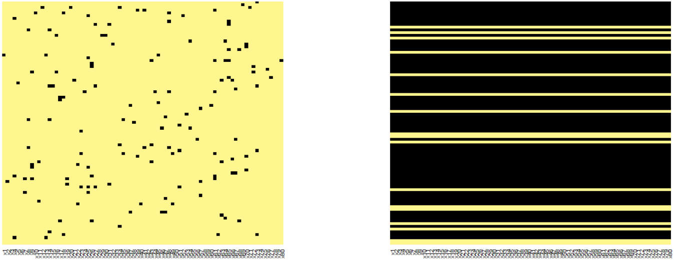


Figure: $n = 100 \times p = 80$ data-matrix (left) with a 2% of outlying cells and trimmed x_i with casewise trimming in black (right)

Cellwise trimming

- **Cellwise trimming-TCLUST** [Zaccaria, *G-E*, Greselin and Mayo-Iscar 2025+]

Cellwise trimming

- **Cellwise trimming-TCLUST** [Zaccaria, *G-E*, Greselin and Mayo-Iscar 2025+]
  - Extension to mixture modelling of the cellMCD [Raymaekers and Rousseuw 2024]

Cellwise trimming

- **Cellwise trimming-TCLUST** [Zaccaria, *G-E*, Greselin and Mayo-Iscar 2025+]
  - ◇ Extension to mixture modelling of the cellMCD [Raymaekers and Rousseuw 2024]
  - ◇ Alternating steps: Detection of outlying cells ↔ EM for Gaussian mixtures with missing cells

Cellwise trimming

- **Cellwise trimming-TCLUST** [Zaccaria, *G-E*, Greselin and Mayo-Iscar 2025+]
  - ◇ Extension to mixture modelling of the cellMCD [Raymaekers and Rousseuw 2024]
  - ◇ Alternating steps: Detection of outlying cells ↔ EM for Gaussian mixtures with missing cells
- **Cellwise trimming-RLG** [*G-E*, Rivera, Mayo-Iscar and Ortega 2021]

Cellwise trimming

- **Cellwise trimming-TCLUST** [Zaccaria, *G-E*, Greselin and Mayo-Iscar 2025+]
  - Extension to mixture modelling of the cellMCD [Raymaekers and Rousseuw 2024]
  - Alternating steps: Detection of outlying cells ↔ EM for Gaussian mixtures with missing cells
- **Cellwise trimming-RLG** [*G-E*, Rivera, Mayo-Iscar and Ortega 2021]
  - Extension of the cellwise-robust PCA [Maronna and Yohai 2012]

Cellwise trimming

- **Cellwise trimming-TCLUST** [Zaccaria, *G-E*, Greselin and Mayo-Iscar 2025+]
  - Extension to mixture modelling of the cellMCD [Raymaekers and Rousseuw 2024]
  - Alternating steps: Detection of outlying cells $\leftrightarrow$ EM for Gaussian mixtures with missing cells
- **Cellwise trimming-RLG** [*G-E*, Rivera, Mayo-Iscar and Ortega 2021]
  - Extension of the cellwise-robust PCA [Maronna and Yohai 2012]
  - Alternating Weighted Least Squares (AWLS)

Cellwise trimming and tHDDC

1. Given $\{R_0, R_1, ..., R_G\}$, apply AWLS or MacroPCA [Hubert, Rousseeuw and van den Bossche 2019] to update affine subspaces $\mathcal{B}_g$ and eigenvalues for $\{x_i : i \in R_g\}$

Cellwise trimming and tHDDC

1. Given $\{R_0, R_1, ..., R_G\}$, apply AWLS or MacroPCA [Hubert, Rousseeuw and van den Bossche 2019] to update affine subspaces $\mathcal{B}_g$ and eigenvalues for $\{x_i : i \in R_g\}$
2. $\Pr_{\mathcal{B}_g}(x_i)$ is needed to update $\{R_0, R_1, ..., R_G\}$ $\rightsquigarrow$ Not straightforward for $\widetilde{g} \neq g$ if $i \in R_g$ (unknown positions of "unreliable" cells...)

Cellwise trimming and tHDDC

1. Given $\{R_0, R_1, ..., R_G\}$, apply AWLS or MacroPCA [Hubert, Rousseeuw and van den Bossche 2019] to update affine subspaces $\mathcal{B}_g$ and eigenvalues for $\{x_i : i \in R_g\}$
2. $\Pr_{\mathcal{B}_g}(x_i)$ is needed to update $\{R_0, R_1, ..., R_G\}$ $\rightsquigarrow$ Not straightforward for $\widetilde{g} \neq g$ if $i \in R_g$ (unknown positions of "unreliable" cells...)
3. LTS-based predictions for $\Pr_{\mathcal{B}_{\widetilde{g}}}(x_i)$ [*G-E*, Rivera, Mayo-Iscar and Ortega 2021]

Cellwise trimming and tHDDC

1. Given $\{R_0, R_1, ..., R_G\}$, apply AWLS or MacroPCA [Hubert, Rousseeuw and van den Bossche 2019] to update affine subspaces $\mathcal{B}_g$ and eigenvalues for $\{x_i : i \in R_g\}$
2. $\Pr_{\mathcal{B}_g}(x_i)$ is needed to update $\{R_0, R_1, ..., R_G\}$ $\rightsquigarrow$ Not straightforward for $\widetilde{g} \neq g$ if $i \in R_g$ (unknown positions of "unreliable" cells...)
3. LTS-based predictions for $\Pr_{\mathcal{B}_{\widetilde{g}}}(x_i)$ [*G-E*, Rivera, Mayo-Iscar and Ortega 2021]

- Computationally intensive approach

Nonlinearity and sparsity

- Address **nonlinearity** through **kernelized** versions [Bouveyron, Fauvel and Girard 2013]

Nonlinearity and sparsity

- Address **nonlinearity** through **kernelized** versions [Bouveyron, Fauvel and Girard 2013]
- Robust **functional clustering** through functional subspaces [Bouveyron and Jacques 2011]

Nonlinearity and sparsity

- Address **nonlinearity** through **kernelized** versions [Bouveyron, Fauvel and Girard 2013]
- Robust **functional clustering** through functional subspaces [Bouveyron and Jacques 2011]
- We just focus on **moderately** (recall the title of the talk!!) high dimensional cases

Nonlinearity and sparsity

- Address **nonlinearity** through **kernelized** versions [Bouveyron, Fauvel and Girard 2013]
- Robust **functional clustering** through functional subspaces [Bouveyron and Jacques 2011]
- We just focus on **moderately** (recall the title of the talk!!) high dimensional cases
  - ◇ **Noise variables** that do not provide useful information about the clustering structure

Nonlinearity and sparsity

- Address **nonlinearity** through **kernelized** versions [Bouveyron, Fauvel and Girard 2013]
- Robust **functional clustering** through functional subspaces [Bouveyron and Jacques 2011]
- We just focus on **moderately** (recall the title of the talk!!) high dimensional cases
  - **Noise variables** that do not provide useful information about the clustering structure
  - **Variable selection** in robust clustering [Ritter 2014] and **sparsity-based** approaches [Kondo, Salibian-Barrera and Zamar 2016; Brodinova et al 2019; Raymaekers and Zamar 2022]

Conclusions

Conclusions and further directions

1. Robust clustering in **high-dimensional** data is of interest

Conclusions and further directions

1. Robust clustering in **high-dimensional** data is of interest
2. TCLUST faces limitations in high-dimensional settings

Conclusions and further directions

1. Robust clustering in **high-dimensional** data is of interest
2. TCLUST faces limitations in high-dimensional settings
3. Proposed method **tHDDC**, as a compromise between TCLUST and RLG, with initial promising results

Conclusions and further directions

1. Robust clustering in **high-dimensional** data is of interest
2. TCLUST faces limitations in high-dimensional settings
3. Proposed method **tHDDC**, as a compromise between TCLUST and RLG, with initial promising results
4. Plenty of room for further research

Universidad de Valladolid

Thanks for your attention!!!!