Supplementary materials: Outlier detection in multivariate functional data through a contaminated mixture model

Martial AMOVIN-ASSAGBA[a,b], Irène GANNAZ[c], Julien JACQUES[a]

[a] *Univ Lyon, Univ Lyon 2, ERIC UR3083, Lyon, France*
[b] *Arpege Master K, Saint-Priest, 69800, France*
[c] *Univ Lyon, INSA Lyon, UJM, UCBL, ECL, ICJ, UMR5208, Villeurbanne, 69621, France*

We present in this document, some additional graphs and comments to our paper titled: *Outlier detection in multivariate functional data through a contaminated mixture model*

1. Simulated data

1.1. Log-likelihood plots for simulated data

The number of clusters is fixed to $K = 4$. $d_k = 2$, for all clusters As an illustration of the behaviours of the algorithm for the three types of initialization, Figure 1 plots the log-likelihoods for one simulation of *dataset1*. This execution illustrates the importance of the initialization strategy, since the achieved maximum is not the same (*trimmed* leads to the highest log-likelihood, *random* to the lowest). Moreover, correctly initializing reduces the number of EM iterations and, consequently, the computation time: if the convergence is achieved with 6 iterations for *trimmed* and *kmeans*, it requires 70 iterations for *random* initialization.

1.2. Application of the algorithm to the simulated data

In order to illustrate the phenomenon that the algorithm tends to group the outliers into additional clusters in section 4.2.3, we display the results on single simulations. First, Figure 2 shows the clusters obtained on one simulation when BIC chooses $K = 4$. The outliers are correctly detected, and are associated to the closest clusters.

Figure 3 shows the clusters obtained on one simulation when BIC chooses $K = 5$. It confirms that the outliers are grouped together into an additional cluster.

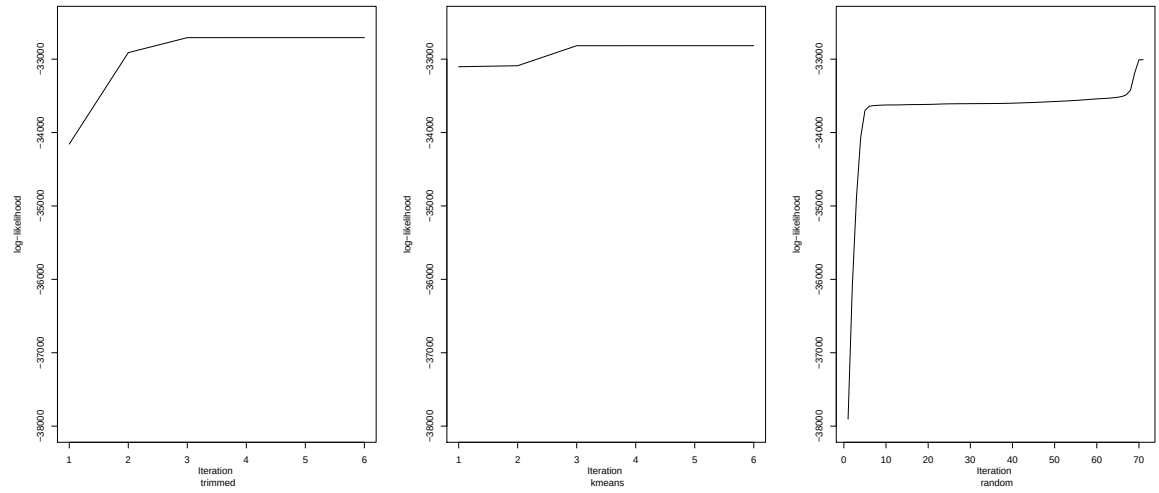

Figure 1: Log-likelihood curves for one simulation of *dataset1* with the three types of initialization.

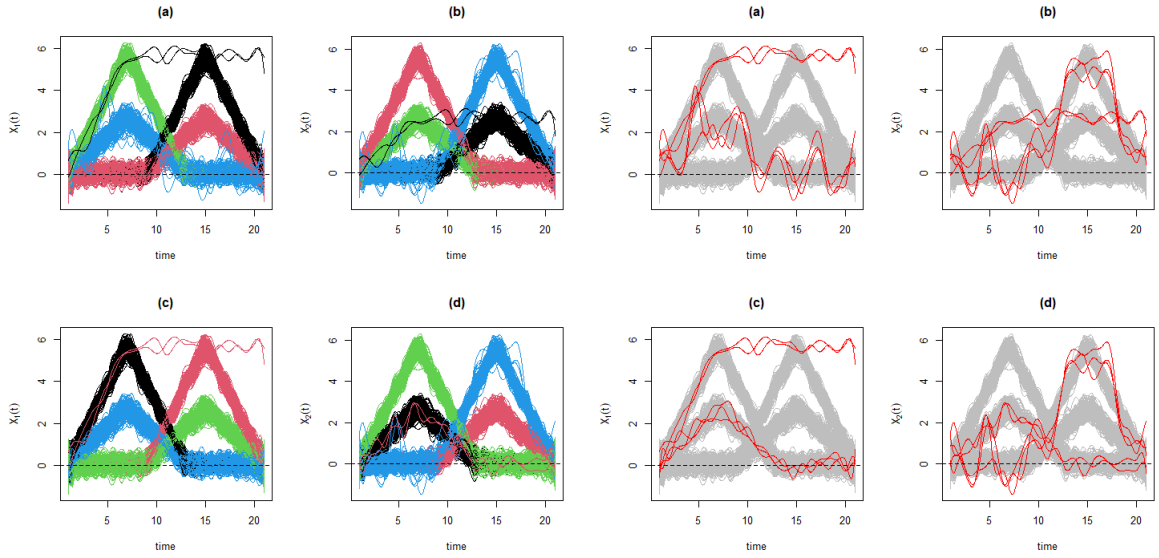

Figure 2: Four left plots: clusters obtained by C-funHDD (by color) when $K = 4$. Four right plots: normal curves are displayed in grey and detected outliers are in red. Top plots give results for *dataset1*: (a) for the first component and (b) for the second. Bottom plots give results for *dataset2*: (c) for the first component and (d) for the second.

1.3. Dealing with funHDDC

We investigate the results of the funHDDC [1] method, a non-robust clustering method. We consider both simulated datasets. The initialization of this method is done with

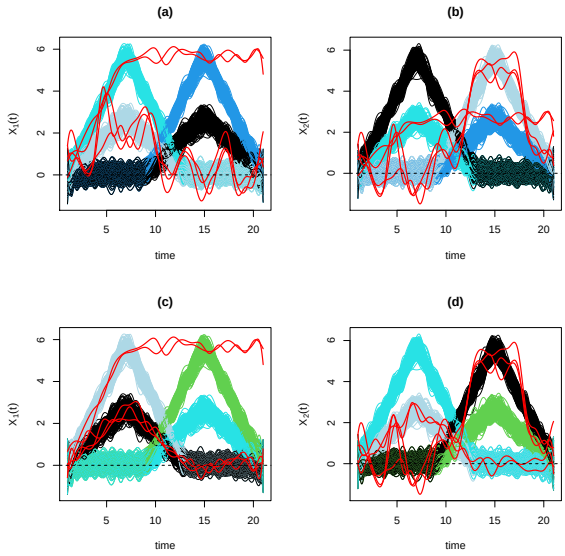

Figure 3: Clusters obtained by C-funHDD (by color) when $K = 5$. Top plots give results for *dataset1*: (a) for the first component and (b) for the second. Bottom plots give results for *dataset2*: (c) for the first component and (d) for the second.

kmeans and *trimmed*. We add the *trimmed* initialization in the algorithm implemented by the authors of funHDDC [1]. The goal is to see if we get a cluster that only contains abnormal data. We vary K from ~~2~~ 1 to 6 with step 1. The values of the intrinsic dimension d_k are fixed at 2 per cluster as in the previous experiments. The results are reported in Table 1.

kmeans			*trimmed*		
(K)	*dataset1*	*dataset2*	(K)	*dataset1*	*dataset2*
2	8	6	2	26	23
3	2	1	3	9	4
4	14	11	**4**	**42**	**45**
5	29	29	5	14	9
6	**47**	**53**	6	9	19

Table 1: funHDDC: best number of clusters selected by the BIC ~~criterion~~ for 100 simulations for each data set as a percentage, with initialization with *kmeans* and *trimmed*. The choice of d_k is undertaken with grid search method. The highest proportions are displayed in bold type.

On the simulated datasets, funHDDC does not correctly choose the number of clusters.

It does not succeed to create a cluster that only contains the abnormal data even though $K = 5$ or $K = 6$ is chosen as the number of clusters. With *trimmed* initialization, it does not recover in general more than four clusters. This highlights the efficiency of having a robust approach.

2. Functional boxplots for the clusters obtained on real data

The two false positives detected by C-funHDDC in our paper are presented in Figure 4.

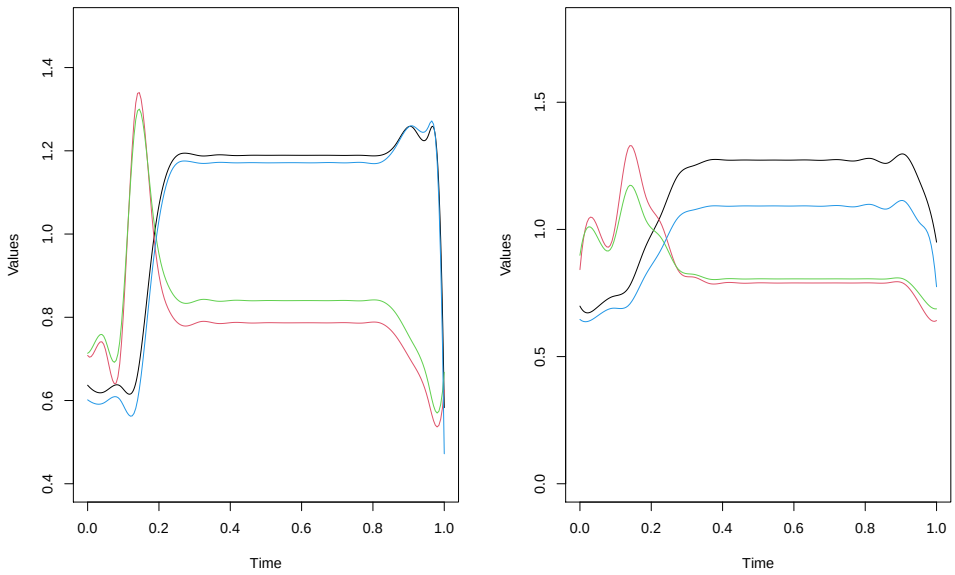

Figure 4: False positive curves normalized in time and amplitude

Figure 5 displays the functional boxplot based on the band depth concept [2] for the two first clusters of normal curves. It is done by the function `fbplot` of the package `fda` in `R`.

The band in magenta is delimited by the 50% deepest curves. The border of this band is defined as the envelope representing the box in a classical boxplot. The blue curves denote these envelopes, and a black curve represents the median curve. The functional boxplot also detect outliers. The red dashed curves are the outliers identify by the functional boxplot. But these curves are not really outliers. These curves represent either a later increase in the values of the sensors or an earlier decrease in the values compared to the others.

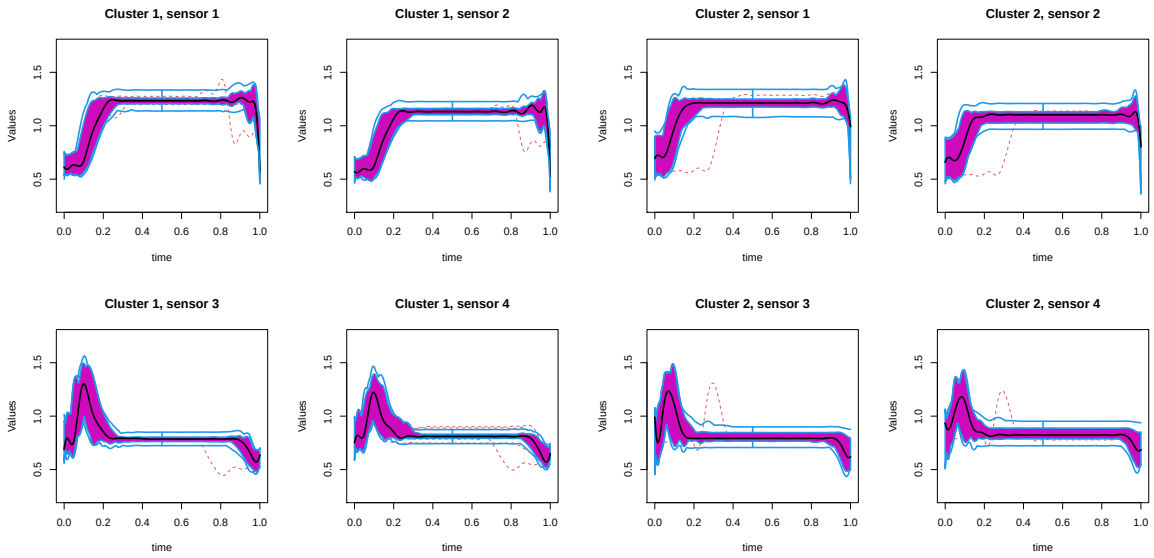

Figure 5: Four left plots: functional boxplots for Cluster 1 (normal data), one boxplot per sensor. Four right plots: functional boxplots for Cluster 2 (normal data), one boxplot per sensor

The first cluster of normal data (cluster 1) differs from the second (cluster 2) when stabilization (plateau) begins. The curves of cluster 2 stabilize faster than the curves of cluster 1. We represent in Figure 6 the observations of cluster 3, which is the cluster of outliers, without the 2 false positives. We also represent the corresponding functional boxplots. It is clear that these boxplots are very different from those of normal data clusters. The outliers are also very different from each other.

References

[1] A. Schmutz, J. Jacques, C. Bouveyron, L. Cheze, P. Martin, Clustering multivariate functional data in group-specific functional subspaces, Computational Statistics (2020) 1–31.

[2] Y. Sun, M. G. Genton, Functional boxplots, Journal of Computational and Graphical Statistics 20 (2) (2011) 316–334.

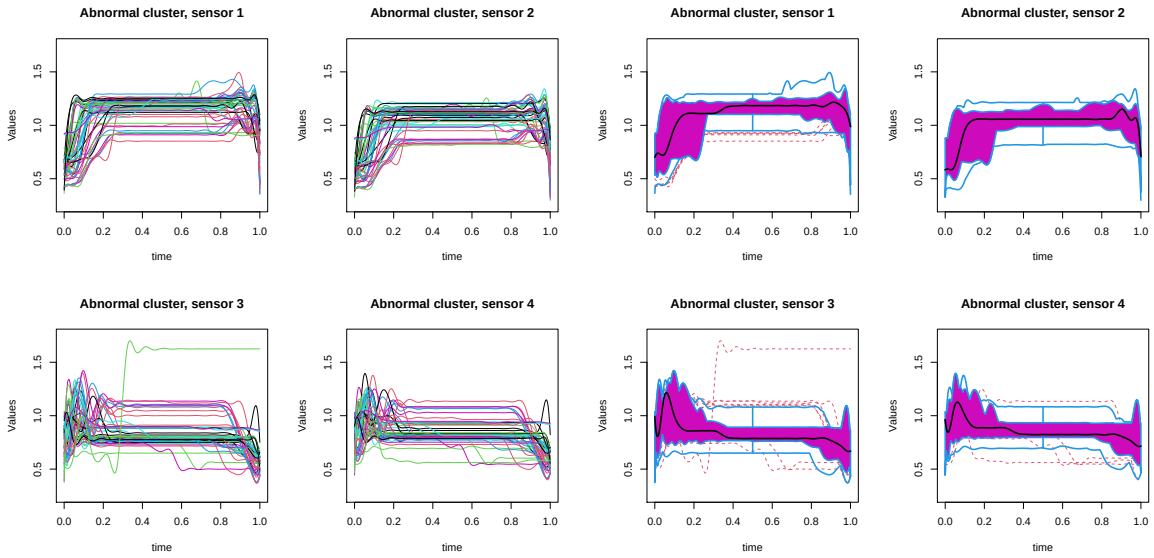

Figure 6: Observations of Cluster 3 (96% of outliers), one plot per sensor.