Final simulation results

Annachiara Rossi

8/31/2022

Simulation setting

A simulation study has been carried out to illustrate how the choice of the covariance estimator for the generation of a clean population from the original, corrupted, data in the functional boxplot can affect the functional outlier detection performance. Such estimators are split into two categories:

1. Multivariate Scatter Estimators:
    - Ledoit-Wolf, the classical regularized covariance operator, as a non-robust benchmark;
    - OGK, a robust estimator used in the current *roahd::fbplot* implementation, which however doesn't guarantee that the scatter matrix is well-conditioned. As univariate robust scale estimators, both MAD and Qn are used.
    - MRCD, a regularized version of MCD, where the covariance matrix is a convex combination of a target matrix and the sample covariance matrix of the h-subset. It is well conditioned by construction. Dealing with functional data, the target matrix has been set to favor an equicorrelation structure. Two values of the percentage of good observations α, 50% and 75%, have been compared, to see how this hyperparameter affects the tuning of the inflation factor F.
    - kernel-MRCD, in the version with linear kernel, for a comparison with MRCD implementation. It includes a correction by a consistency factor and a refinement step, unlike MRCD. Furthermore, the target matrix is fixed to the identity. These differences will be commented in the results' section. The advantage of the kernel version over the standard one should be found in the computational time, since our data counts more dimensions than cases.
2. Functional Covariation Estimators, which make use of the functional nature of data. Their estimates have the same eigenfunctions as the sample covariance operator, but not the same eigenvalues. The latter will be estimated as the variance of data projected over the eigenfunctions. They are:
    - Spherical Covariance, the covariance of data projected onto a unit sphere with center in the spatial median;
    - Median Covariation, a median-type operator for dispersion;
    - Kendall's τ function, which doesn't give a Scatter estimate, but has has the same eigenspace as the population covariance operator.

In the functional boxplot implementation, two distinct simulation criteria are used. For multivariate estimators, the function $roahd :: generate\_gauss\_fdata$ exploits the Cholesky decomposition of the computed Scatter matrix to generate samples. For functional estimators, a KL-type generative model is employed, which makes use of the estimates of the eigenspace.

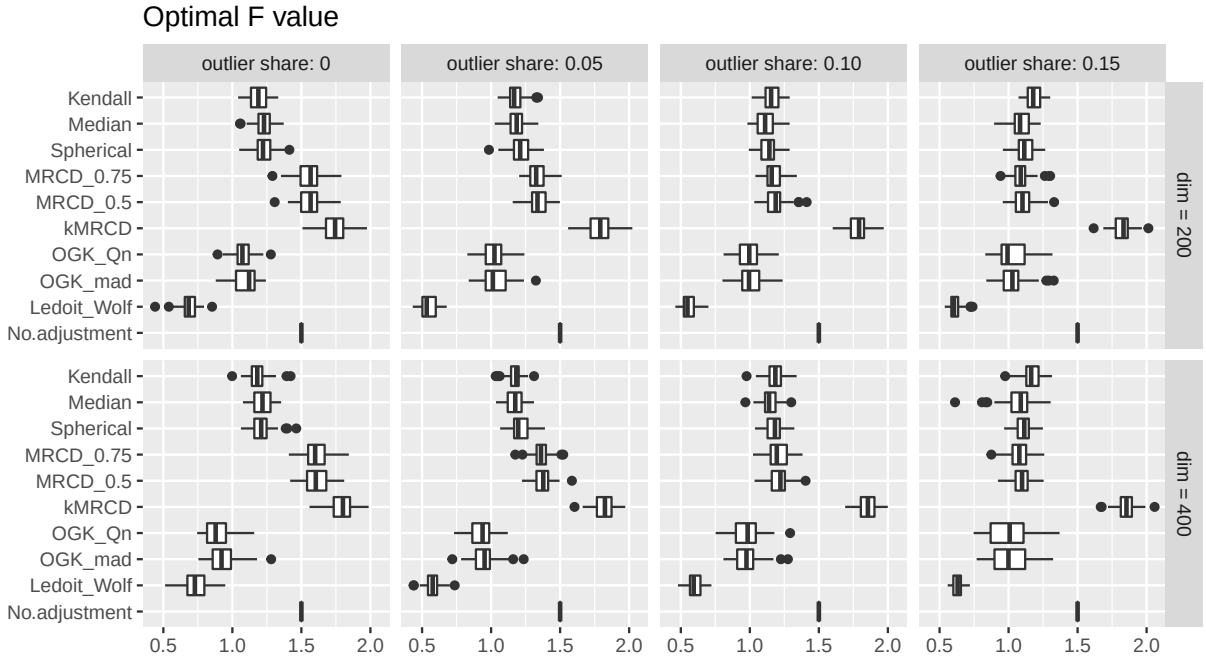
Data has been generated with 100 samples, and two possible dimensionalities $p = \{200, 400\}$. Since we need to exploit the functional features of data, to make use of the second class of operators, we started constructing a spectrum, built using an Exponential Covariance matrix, and a basis composed of eigenfunctions. The generating process is Gaussian, with $\underline{0}$ mean and covariance matrix having the eigenvalues on its diagonal.

From this, a functional datum can be constructed, adding a mean sine process. Such data should now be contaminated. For each data dimensionality, four are the proportions of outlying curves over the population which have been treated: 0% - no contamination, $5\%, 10\%, 15\%$. The corrupted curves have been inflated, to obtain some simple **amplitude outliers**.

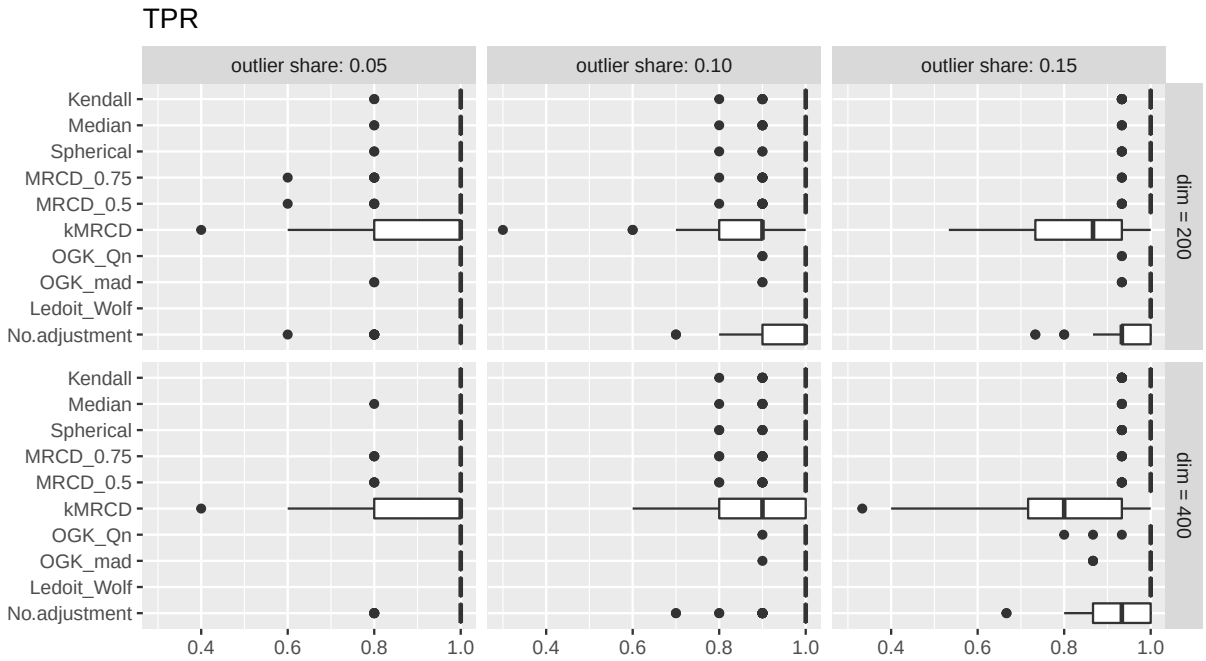
Simulation results

The simulation has been performed on the HPC system provided by the MOX office, over the gigatlong queue. It puts at our disposal 32 cores and 252gb of RAM. For each combination of the parameters above described, $B = 64$ repetitions have been carried out in parallel by the 32 cores. Overall, the simulation took approximately five hours.

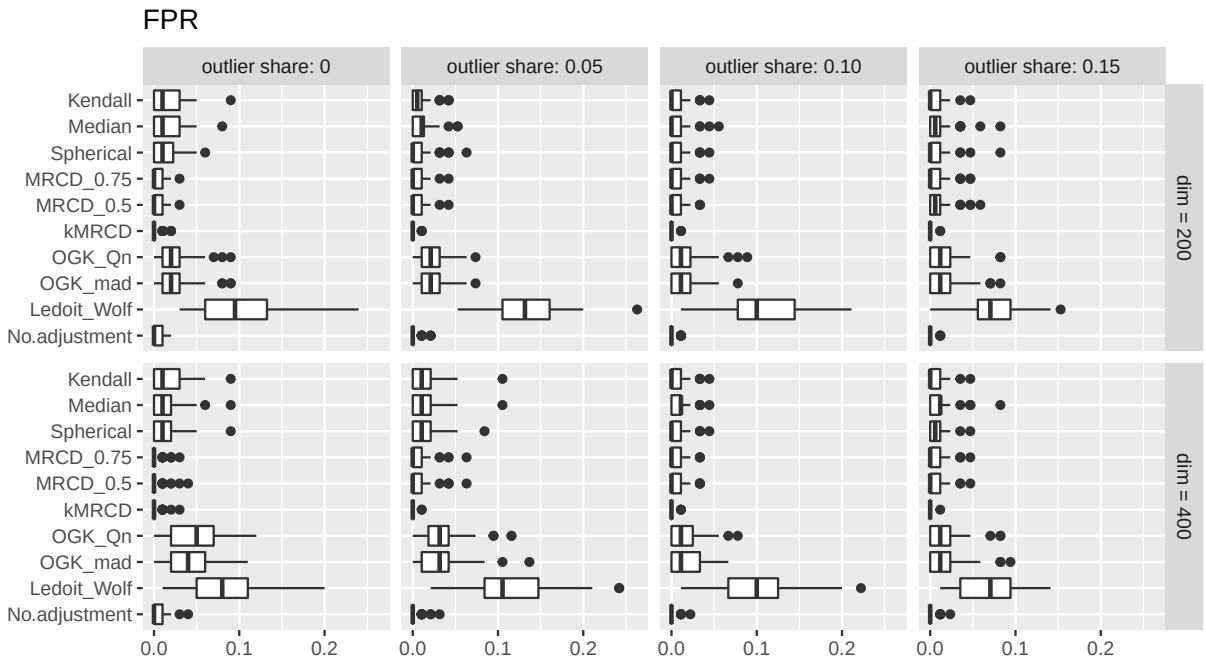
In the following, the distributions of the tuned inflation factor and some metrics of interest will be shown.



We can see that functional operators behave similarly: Kendall's τ function has a tendency to lower values of F for smaller outliers proportions, and higher F^* for higher outlier shares, with respect to Median and Spherical. MRCD and kMRCD happen to lead to $F^* > 1.5$, while OGK and Ledoit-Wolf are always under the default value of 1.5 and sometimes very small. This behavior will lead to many false positives, i.e., curves wrongly flagged as outlying.

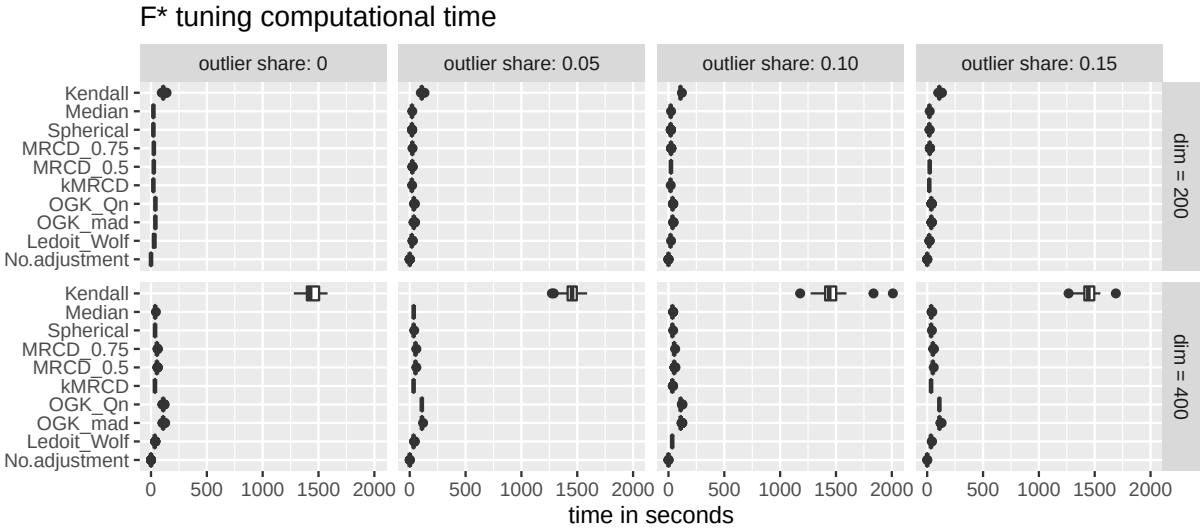


The True Positive rate gives out a satisfying performance by the majority of the estimators, apart from kMRCD, whose distribution reveals some difficulty in flagging the samples: many outlying observations are not spotted. This is coherent with the high values of F^* obtained in the previous plot. Regarding the case in which F^* is fixed to 1.5, we notice an increasing difficulty in correctly identifying all outliers as their proportion increases, which justifies the need for an adjustment of the inflation factor based on data itself.



The False Positive rate ranges to a maximum of 0.3, and has an elongated distribution for the non-robust benchmark, as we would have expected. We can see that the currently employed estimator in the functional boxplot, OGK, also presents a more dispersed distribution with respect to the newly presented ones, which

give a higher precision. The kernel version of MRCD is the best-behaving in this framework, but, as we have seen before, has many troubles in identifying the inflated samples.



The computational time required for the adjustment of the inflation factor is comparable among all the estimators, but Kendall's τ function. Indeed, to get the estimate of the eigenfunctions from the latter, high computational power and resources are required. A zoom over the remaining estimators is shown in the following plot, which highlights some interesting differences. OGK is in any case the slowest among all methods. As we were expecting, kMRCD is faster than the non-kernelized version, despite the additional computation due to the refinement step. This gap is more evident in higher dimensionality. Overall, the Spherical and Median Covariation Operators are the most efficient, as they are the least affected by the doubled dimensionality of the data.