TFunHDDC Python Test Results

TFunHDDC was run with default settings in Python and R, except for K = 2 and nb\_rep = 20. Data used was NOxBenchmark data. Each initialization was run 100 times.

|  |  |  |
| --- | --- | --- |
| Init | Avg. CCR (stdev) | Avg. ARI (stdev) |
| Random (Py) | 0.502522  (0.018458) | -0.04838  (0.002858) |
| Random (R) | 0.501826  (0.015842) | -0.04979  (0.005236) |
| Vector (Py) | 0.513043  (0) | -0.05031  (0) |
| Vector (R) | 0.513043  (0) | -0.05031  (0) |
| Mini-EM (Py) | 0.49913  (0.01824) | -0.04881  (0.005358) |
| Mini-EM (R) | 0.498348  (0.026528) | -0.04775  (0.025642) |
| Kmeans (Py) | 0.499217  (0.01302) | -0.05031  (0) |
| Kmeans (R) | 0.501739  (0.045755) | -0.04134  (0.052331) |

Overall the results seem very consistent between the R version and the Python version. They seem to correctly classify the data similarly (or at the very least make a similar number of mistakes when classifying), and both seem to keep a consistent ARI around -0.05.

Interestingly, the Python version generally had a smaller standard deviation, maybe suggesting that the results were more consistent between runs. However, this may be due to the presence of a small number of outliers in the R runs.

Further testing should be done to see if the two versions will pick the same parameter combinations when given multiple (eg. K=[2,3,4,…], threshold=[0.1,0.01, …], etc…).

See next page for boxplots of the 100 runs in R and Python.

A graph with numbers and a line

Description automatically generated with medium confidencePython Boxplot (1 is random, 2 is vector, 3 is mini-em, and 4 is kmeans)

A graph with black lines

Description automatically generatedR Boxplot