Detect 1 change point for one-dimensional data using DISCO measure

Suppose our observed data $\boldsymbol{Z}$ is composed by

$$X \sim F_X$$
$$Y \sim F_Y$$

for some distinct distributions F_X and F_Y with their corresponding characteristic functions ϕ_X and ϕ_Y respectively. Suppose we observe $\boldsymbol{Z} = (Z_1, ..., Z_T) = (X_1, ..., X_{N_1}, Y_1, ..., Y_{N_2})$, where $N_1 + N_2 = T$. Our goal is to estimate the change point c such that $Z_1, ..., Z_c \sim F_X$ and $Z_{c+1}, ..., Z_T \sim F_y$.

Suppose there is a rolling window of width d. We can generate a series of rolling observations $W_1, ..., W_n$ where $W_i = (Z_i, .., Z_{i+d-1})$. For each W_i there is an associated characteristic function ϕ_i for $Z_i, .., Z_{i+d-1}$. If each Z_i in this rolling window comes from the same distribution, e.g. $Z_i, ..., Z_{i+d-1} \sim F_X$, then $\phi_i = \phi_X$. We hope that by clustering characteristic functions we can find the change point in $\boldsymbol{Z}$.

Let Φ be the space of characteristic functions, with the measure of distance defined as

$$d(\phi_i, \phi_j) = \int_{\mathbb{R}^d} |\phi_i(t) - \phi_j(t)|^2 \omega(t) dt.$$

Szekely and Rizzo [2005] show that this measure of distance can be approximated by

$$\varepsilon(W_i, W_j, \alpha).$$

If we have a reasonably large window size, we should expect the estimated distance between ϕ_i and either ϕ_X or ϕ_Y to be small when W_i is homogeneous.

Matteson and James [2014] propose an agglomerative algorithm which can be viewed as a hierarchical clustering method. We want go one step further by suggesting that we can apply any plausible clustering methods here to estimate number of change points and location of change points in one step. The challenge is that we only know the approximation of the metric for the space of characteristic functions Φ. Other things, such as the mean or the variance are not well defined. Therefore, many of the commonly used clustering methods, such as the traditional K-means, cannot be applied directly. We need to find clustering methods that only require the pairwise distance.

The one I am trying now is called the self-organizing maps, which is a competitive learning algorithm that is commonly used in image process. Briefly speaking, we start with K connected nodes in the space. Every time we randomly pick one point from observation, and calculate the distances between this chosen point and all nodes. The closest one is called the winning node. It will move toward the chosen point for some distance, while the two nodes beside this winning node will also move a little bit. As we can see, this algorithm only requires calculating the distances. I tried this method with the simplest case of one change point and two nodes, and the detailed steps are described as follows.

1. Randomly choose two rolling windows from $W_1, ..., W_n$ as the starting nodes C_1 and C_2.

2. Randomly choose a rolling window $S = (S_1, ..., S_d) \in (Z_1, ..., Z_T)$ from $W_1, ..., W_n$, calculate the distance $d(S, C_1)$ and $d(S, C_2)$. The node with smaller distance is denoted

as C_w with the corresponding observations $(P_1, ..., P_d) \in (Z_1, ..., Z_T)$.

3. Construct the new node by conduction a random sampling with $p\%$ observations from S and $1 - p\%$ from C_w. The new node is denoted as C'. It can be shown that $d(S, C') < d(S, C_w)$, and this is how we mimic the "moving towards the chosen point" in the space Φ.

4. Repeat Step 2-3 K times. Then assign the cluster by the distances between each point and the two nodes.

5. Apply the agglomerative algorithm to a small neighborhood of the boundary of the two clusters.

Right now the self-organizing maps algorithm does not outperforms the method proposed by Matteson and James [2014]. But I would expect our new method to work faster when there are multiple change points and T is large. I did a benchmark experiment with $X \sim N(0,1)$, $Y \sim N(1,1)$, and $T = 10^4$. Our method could be at least 20 times faster then the method proposed by Matteson and James [2014] with basically the same results. When $T = 2 \times 10^4$ our method is about 100 times faster. When $T = 10^5$ the other method fails immediately since it requires about 40 Gb of memory which is not feasible, while our method is not really affected by the large data size.

Table 1: $T = 3000$, $N_1 = 1000$, $N_2 = 2000$, and $X \sim N(0,1)$.

	$N(1,1)$	$N(0,2)$	$t(2)$	$t(16)$
MJ	$0.995_{(.006)}$	$0.992_{(.011)}$	$0.975_{(.039)}$	$0.322_{(.244)}$
New Method	$0.992_{(.043)}$	$0.977_{(.090)}$	$0.816_{(.227)}$	$0.432_{(.327)}$

My plan for the next step is to

- Make the SOM algorithm work for multiple change points.

- Develop methods for estimating number of change points.
- Explore other clustering methods.

References

David S Matteson and Nicholas A James. A nonparametric approach for multiple change point analysis of multivariate data. *Journal of the American Statistical Association*, 109 (505):334–345, 2014.

Gabor J Szekely and Maria L Rizzo. Hierarchical clustering via joint between-within distances: Extending ward's minimum variance method. *Journal of classification*, 22(2): 151–183, 2005.