Discussion of *Automatic Change-Point Detection in Time Series via Deep Learning* by Li, Fearnhead, Fryzlewicz, and Wang

Shakeel Gavioli-Akilagun[1]

[1]Department of Statistics, London School of Economics

August 2023

We congratulate the authors on the interesting and thought provoking paper. Lemma 3.2 shows that the Generalised Likelihood Ratio Test (GLR) for stability of a linear regression can be viewed as a simple feed forward neural network. However, close inspection of the lemma reveals the setup rules out several common change point problems. Consider the piecewise polynomial regression:

$$y_i = \begin{cases} \sum_{j=0}^{p} \alpha_j \left(i/n - \tau/n\right)^j + \xi_i & \text{if } t \leq \tau \\ \sum_{j=0}^{p} \beta_j \left(i/n - \tau/n\right)^j + \xi_i & \text{if } t > \tau \end{cases} \qquad i = 1, \ldots, n. \tag{1}$$

For ξ's distributed i.i.d. $\mathcal{N}(0,1)$ the likelihood ratio statistic for a change at location i is

$$\mathcal{R}_i(\mathbf{Y}) = \|P_{1:i}\mathbf{Y}\|_2 + \left\|P_{(i+1):n}\mathbf{Y}\right\|_2 - \|P_{1:n}\mathbf{Y}\|_2 \tag{2}$$

where $P_{s:e}\mathbf{Y}$ denotes the projection of elements indexed $\{s, \ldots, e\}$ in the vector $\mathbf{Y} = (y_1, \ldots, y_n)'$ onto the space of (discretised) polynomials of degree p. Since (2) is a linear combination of quadratic forms $h_\lambda^{\mathrm{GLR}}(\boldsymbol{y}) = \mathbf{1}_{\{\max_i \mathcal{R}_i(\boldsymbol{y}) > \lambda\}}$ clearly cannot be represented as a neural network. The Wald test for the same problem (e.g. [KOC22]) likewise cannot be represented in this way.

In [GAF23] we introduce tests based on differences of local sums of the data as simple and computationally efficient alternatives to GLR and Wald tests. Interestingly, our difference based tests can be represented as a neural network. Consider the statistic

$$\mathcal{D}_i\left(\mathbf{Y}\right) = \left\{\sum_{j=0}^{p+1}\binom{p+1}{j}^2\right\}^{-1/2}\sum_{k=0}^{p+1}(-1)^{p+1-k}\binom{p+1}{k}\left[\frac{y_{i+(k-1)l_i+1}+\cdots+y_{i+kl_i}}{\sqrt{l_i}}\right]$$

where $l_i = \max\left\{l \in \mathbb{Z} \mid i-l \geq 0 \text{ and } i+(p+1)l \leq n\right\}$. Since $\mathcal{D}\left(\cdot\right)$ is a linear operator $h_\lambda^{\mathrm{DIF}}\left(\boldsymbol{x}\right) = \mathbf{1}_{\{\max_i|\mathcal{D}_i(\boldsymbol{x})|>\lambda\}}$ can be represented as a neural network. Using the techniques in [GAF23] one can show that the localization rate of Algorithm 1 for the change point in (1) is of the order

$$\mathcal{O}\left(\frac{B^2 n^{\frac{2p^*}{2p^*+1}}}{\Delta_{p^*}^2}\right)$$

where $\Delta_j = (\alpha_j - \beta_j)$, $p^* \in \operatorname{argmax}_{j=0,\ldots,p}\left\{|\Delta_j|\left(\delta/n\right)^j\right\}$, and $\delta = \tau \wedge (n-\tau)$. This is unimprovable up to the B^2 term. When analysing the behaviour of neural networks on change point problems it may be useful to think in terms of difference based tests.

References

[GAF23] Shakeel Gavioli-Akilagun and Piotr Fryzlewicz. Fast and optimal inference for change points in piecewise polynomials via differencing. *arXiv preprint arXiv:2307.03639*, 2023.

[KOC22] Joonpyo Kim, Hee-Seok Oh, and Haeran Cho. Moving sum procedure for change point detection under piecewise linearity. *arXiv preprint arXiv:2208.04900*, 2022.