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

Shakeel Gavioli-Akilagun

London School of Economics
Department of Statistics

August 2023

THE LONDON SCHOOL OF ECONOMICS AND POLITICAL SCIENCE

- A key idea is that common change point tests can be represented as single layer feed forward neural networks with RELU activation.

Lemma (3.2)

Consider the change point model:

$$y_i = \beta' \mathbf{z}_i + \phi c_{\tau,i} + \xi_i \qquad i = 1, \ldots, n$$

Where $c_{\tau,i}$ is a scalar covariate specific to the change at τ and $\xi_i \sim \mathcal{N}\left(0, \sigma^2\right)$. Then there is an $h^* \in \mathcal{H}_{1,2n-2}$ equivalent to the likelihood-ratio test for testing $\phi = 0$ against $\phi \neq 0$.

- Apparently, the setup rules out several common change point problems.

- A key idea is that common change point tests can be represented as single layer feed forward neural networks with RELU activation.

Lemma (3.2)

Consider the change point model:

$$y_i = \boldsymbol{\beta}'\mathbf{z}_i + \phi c_{\tau,i} + \xi_i \qquad i = 1, \ldots, n$$

Where $c_{\tau,i}$ is a scalar covariate specific to the change at τ and $\xi_i \sim \mathcal{N}\left(0, \sigma^2\right)$. Then there is an $h^* \in \mathcal{H}_{1,2n-2}$ equivalent to the likelihood-ratio test for testing $\phi = 0$ against $\phi \neq 0$.

- Apparently, the setup rules out several common change point problems.

- A key idea is that common change point tests can be represented as single layer feed forward neural networks with RELU activation.

Lemma (3.2)

Consider the change point model:

$$y_i = \boldsymbol{\beta}' \mathbf{z}_i + \phi c_{\tau,i} + \xi_i \qquad i = 1, \ldots, n$$

Where $c_{\tau,i}$ is a scalar covariate specific to the change at τ and $\xi_i \sim \mathcal{N}(0, \sigma^2)$. Then there is an $h^* \in \mathcal{H}_{1,2n-2}$ equivalent to the likelihood-ratio test for testing $\phi = 0$ against $\phi \neq 0$.

- Apparently, the setup rules out several common change point problems.

- Consider the piecewise polynomial change point model

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

- For ξ's distributed i.i.d. $\mathcal{N}(0,1)$ the likelihood ratio statistic (e.g. [BCF19]) for a change at location i is: $\mathcal{R}_i(\mathbf{Y}) = \|P_{1:i}\mathbf{Y}\|_2 + \left\|P_{(i+1):n}\mathbf{Y}\right\|_2 - \|P_{1:n}\mathbf{Y}\|_2$.
- Being 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 single layer neural network with RELU activation.
- The Wald test (e.g. [KOC22]) likewise cannot be represented in this way.
- Natural ways to address this: data pre-processing, different activation functions, etc.

Piecewise polynomials

- Consider the piecewise polynomial change point model

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

- For ξ's distributed i.i.d. $\mathcal{N}(0,1)$ the likelihood ratio statistic (e.g. [BCF19]) for a change at location i is: $\mathcal{R}_i(\mathbf{Y}) = \|P_{1:i}\mathbf{Y}\|_2 + \left\|P_{(i+1):n}\mathbf{Y}\right\|_2 - \|P_{1:n}\mathbf{Y}\|_2$.
- Being 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 single layer neural network with RELU activation.
- The Wald test (e.g. [KOC22]) likewise cannot be represented in this way.
- Natural ways to address this: data pre-processing, different activation functions, etc.

Piecewise polynomials

- Consider the piecewise polynomial change point model

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

- For ξ's distributed i.i.d. $\mathcal{N}(0,1)$ the likelihood ratio statistic (e.g. [BCF19]) for a change at location i is: $\mathcal{R}_i(\mathbf{Y}) = \|P_{1:i}\mathbf{Y}\|_2 + \left\|P_{(i+1):n}\mathbf{Y}\right\|_2 - \|P_{1:n}\mathbf{Y}\|_2$.
- Being 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 single layer neural network with RELU activation.
- The Wald test (e.g. [KOC22]) likewise cannot be represented in this way.
- Natural ways to address this: data pre-processing, different activation functions, etc.

Piecewise polynomials

- Consider the piecewise polynomial change point model

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

- For ξ's distributed i.i.d. $\mathcal{N}(0,1)$ the likelihood ratio statistic (e.g. [BCF19]) for a change at location i is: $\mathcal{R}_i(\mathbf{Y}) = \|P_{1:i}\mathbf{Y}\|_2 + \left\|P_{(i+1):n}\mathbf{Y}\right\|_2 - \|P_{1:n}\mathbf{Y}\|_2$.
- Being 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 single layer neural network with RELU activation.
- The Wald test (e.g. [KOC22]) likewise cannot be represented in this way.
- Natural ways to address this: data pre-processing, different activation functions, etc.

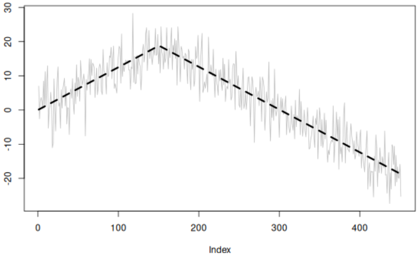
Piecewise polynomials

- Consider the piecewise polynomial change point model

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

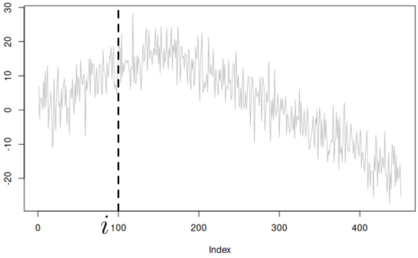
- For ξ's distributed i.i.d. $\mathcal{N}(0,1)$ the likelihood ratio statistic (e.g. [BCF19]) for a change at location i is: $\mathcal{R}_i(\mathbf{Y}) = \|P_{1:i}\mathbf{Y}\|_2 + \left\|P_{(i+1):n}\mathbf{Y}\right\|_2 - \|P_{1:n}\mathbf{Y}\|_2$.
- Being a linear combination of quadratic forms $h_\lambda^{\mathrm{GLR}}(\boldsymbol{y}) = \mathbf{1}_{\{\max_i \mathcal{R}_i(\boldsymbol{y}) > \lambda\}}$ clearly cannot be represented as a single layer neural network with RELU activation.
- The Wald test (e.g. [KOC22]) likewise cannot be represented in this way.
- Natural ways to address this: data pre-processing, different activation functions, etc.

- In [GAF23] we introduce tests based on differences of local sums of the data. Interestingly, our difference based tests can be represented as a neural network.



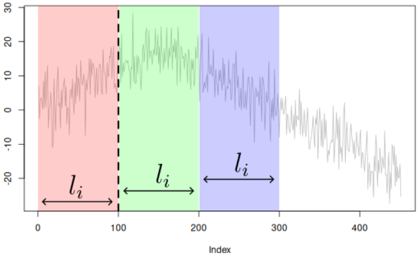
- Since $\mathcal{D}(\cdot)$ is a linear operator $h_\lambda^{\mathrm{DIF}}(\boldsymbol{x}) = \mathbf{1}_{\{\max_i |\mathcal{D}_i(\boldsymbol{x})| > \lambda\}}$ can be represented as a neural network with RELU activation.

- In [GAF23] we introduce tests based on differences of local sums of the data. Interestingly, our difference based tests can be represented as a neural network.



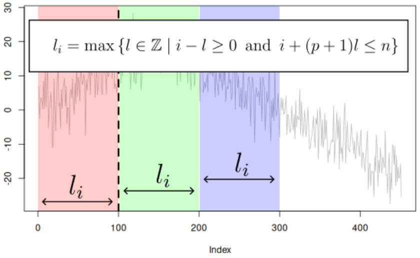
- Since $\mathcal{D}(\cdot)$ is a linear operator $h_\lambda^{\mathrm{DIF}}(\boldsymbol{x}) = \mathbf{1}_{\{\max_i |\mathcal{D}_i(\boldsymbol{x})| > \lambda\}}$ can be represented as a neural network with RELU activation.

- In [GAF23] we introduce tests based on differences of local sums of the data. Interestingly, our difference based tests can be represented as a neural network.



- Since $\mathcal{D}(\cdot)$ is a linear operator $h_\lambda^{\text{DIF}}(\boldsymbol{x}) = \mathbf{1}_{\{\max_i |\mathcal{D}_i(\boldsymbol{x})| > \lambda\}}$ can be represented as a neural network with RELU activation.

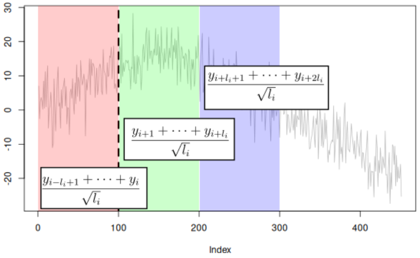
- In [GAF23] we introduce tests based on differences of local sums of the data. Interestingly, our difference based tests can be represented as a neural network.



- Since $\mathcal{D}(\cdot)$ is a linear operator $h_\lambda^{\mathrm{DIF}}(\boldsymbol{x}) = \mathbf{1}_{\{\max_i |\mathcal{D}_i(\boldsymbol{x})| > \lambda\}}$ can be represented as a neural network with RELU activation.

Change point tests based on differencing (I)

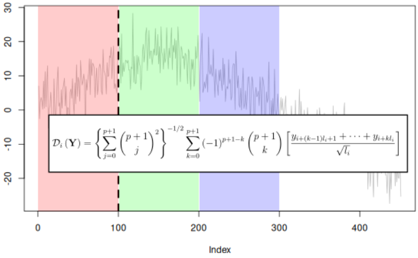
- In [GAF23] we introduce tests based on differences of local sums of the data. Interestingly, our difference based tests can be represented as a neural network.



- Since $\mathcal{D}(\cdot)$ is a linear operator $h_{\lambda}^{\text{DIF}}(\boldsymbol{x}) = \mathbf{1}_{\{\max_i |\mathcal{D}_i(\boldsymbol{x})| > \lambda\}}$ can be represented as a neural network with RELU activation.

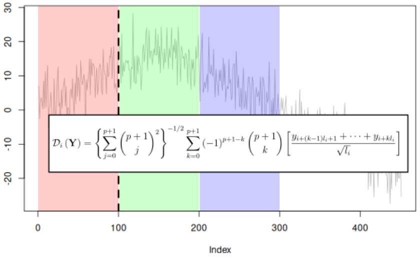
Change point tests based on differencing (I)

- In [GAF23] we introduce tests based on differences of local sums of the data. Interestingly, our difference based tests can be represented as a neural network.



- Since $\mathcal{D}(\cdot)$ is a linear operator $h_\lambda^{\mathrm{DIF}}(\boldsymbol{x}) = \mathbf{1}_{\{\max_i |\mathcal{D}_i(\boldsymbol{x})| > \lambda\}}$ can be represented as a neural network with RELU activation.

- In [GAF23] we introduce tests based on differences of local sums of the data. Interestingly, our difference based tests can be represented as a neural network.



- Since $\mathcal{D}(\cdot)$ is a linear operator $h_\lambda^{\text{DIF}}(\boldsymbol{x}) = \mathbf{1}_{\{\max_i |\mathcal{D}_i(\boldsymbol{x})| > \lambda\}}$ can be represented as a neural network with RELU activation.

Change point tests based on differencing (II)

- Using the techniques in [GAF23] one can show that neural network's localisation rate (Theorem A.6 for Algorithm 1 in the paper) for τ is of the order:

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

Where: $\Delta_j = (\alpha_j - \beta_j)$, $\delta = \tau \wedge (n - \tau)$, $p^* \in \operatorname{argmax}_j \left\{|\Delta_j| \, (\delta/n)^j\right\}$.

- This is unimprovable up to the B^2 term.
- When analyzing the behavior of neural networks on change point problems it may be useful to think in terms of difference based tests.

Change point tests based on differencing (II)

- Using the techniques in [GAF23] one can show that neural network's localisation rate (Theorem A.6 for Algorithm 1 in the paper) for τ is of the order:

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

Where: $\Delta_j = (\alpha_j - \beta_j)$, $\delta = \tau \wedge (n - \tau)$, $p^* \in \text{argmax}_j \left\{ |\Delta_j| \, (\delta/n)^j \right\}$.

- This is unimprovable up to the B^2 term.
- When analyzing the behavior of neural networks on change point problems it may be useful to think in terms of difference based tests.

Change point tests based on differencing (II)

- Using the techniques in [GAF23] one can show that neural network's localisation rate (Theorem A.6 for Algorithm 1 in the paper) for τ is of the order:

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

  Where: $\Delta_j = (\alpha_j - \beta_j)$, $\delta = \tau \wedge (n - \tau)$, $p^* \in \operatorname{argmax}_j \left\{ |\Delta_j| \, (\delta/n)^j \right\}$.

- This is unimprovable up to the B^2 term.
- When analyzing the behavior of neural networks on change point problems it may be useful to think in terms of difference based tests.

References

[BCF19] Rafal Baranowski, Yining Chen, and Piotr Fryzlewicz. Narrowest-over-threshold detection of multiple change points and change-point-like features. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 81(3):649–672, 2019.

[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.