Robust bottom-up algorithms for high dimensional trend segmentation

Hyeyoung Maeng

Durham University, UK

Joint work with Tengyao Wang and Piotr Fryzlewicz at LSE

December 16, 2022

Table of Contents

Introduction

Problem Setup

Our Bottom-up Algorithm
- Wavelet transform
- Thresholding
- Inverse wavelet transform

Theoretical results

Simulation results

Data applications

Summary

Table of Contents

Introduction

Problem Setup

Our Bottom-up Algorithm
- Wavelet transform
- Thresholding
- Inverse wavelet transform

Theoretical results

Simulation results

Data applications

Summary

Recent change point problems

- Single change-point vs Multiple change-point
- Univariate data sequence vs Panel data
- Offline vs Online
  - Offline: A posteriori analysis having observed a series of data. We identify all change-points in a retrospective view.
  - Online: real-time monitoring of a data stream. After observing each new data point, we wish to decide whether or not to flag that a change has occurred.

We focus on offline (posteriori)
multiple change-point detection
for
high dimensional data.

Posteriori multiple change-point detection

Some classes of posteriori multiple change-point detection techniques:

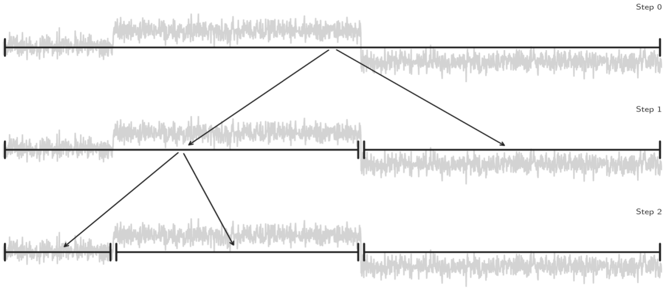
1. **Estimate all change-points at once**
   : Often via minimising a criterion function whose form is likelihood-type fit + penalty term:

$$\underset{\eta_1,\ldots,\eta_N}{\operatorname{argmin}} \Big\{ L(X_t, \eta_1, \ldots, \eta_N) + pen(N, \eta_1, \ldots, \eta_N) \Big\}, \qquad (1)$$

   where $L(\cdot)$ is called loss or cost function that has a form of likelihood (or least-squares) and measures the fit of estimated value to the data, while $pen(\cdot)$ is the penalty added to prevent overfitting.

Posteriori multiple change-point detection

2. **Estimate the change-points one by one** : Well-known example is Binary Segmentation (Vostrikova, 1981) and its variants. The change points are estimated one by one by successively sub-dividing the data in a greedy way (i.e. top-down algorithm).



Posteriori multiple change-point detection

3. **Bottom-up algorithms**: : The natural complement to the Top-down approach but has rarely been used in change-point detection.

- Starting from the finest level of the data and consecutively merging neighbouring segments which are the most likely to correspond to the same feature.
- e.g. The bottom-up adaptive unbalanced Haar wavelet transform for piecewise-constant signal (Fryzlewicz, 2018).

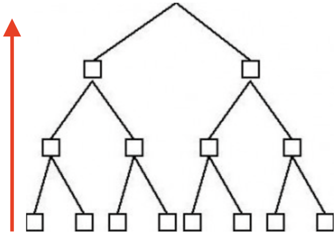


Table of Contents

Introduction

Problem Setup

Our Bottom-up Algorithm
- Wavelet transform
- Thresholding
- Inverse wavelet transform

Theoretical results

Simulation results

Data applications

Summary

High dimensional setting

We consider a change-point model for high-dimensional panel data,

$$X_{i,t} = f_{i,t} + \varepsilon_{i,t}, \quad i = 1, \ldots, n, \ t = 1, \ldots, T$$

where the signal vectors $\{\boldsymbol{f}_i\}_{i=1}^n$ have piecewise-constant structure and share N distinct change-points,

$$0 = \eta_0 < \eta_1 < \eta_2 < \ldots < \eta_N < \eta_{N+1} = T, \tag{2}$$

in that at each change-point η_ℓ, there exists at least one signal $\boldsymbol{f}_i$ in which the trend change occurs at f_{i,η_ℓ}. $\boldsymbol{\varepsilon}_i = (\varepsilon_{i,1}, \ldots, \varepsilon_{i,T})^\top$ is the independent Gaussian random error with $E(\boldsymbol{\varepsilon}_i) = 0$, $\mathrm{Var}(\boldsymbol{\varepsilon}_i) = \sigma_i^2$.

Now the goal is to detect

1. the number N of change points,
2. the locations $\eta_1, \eta_2, \ldots, \eta_N$ of change points.

Dense change vs Sparse change

For each change-point η_ℓ, the change can occur in a dense subset of the coordinates (e.g. all coordinates $\{\boldsymbol{f}_i\}_{i=1}^n$) or only in a sparse subset of the coordinates.

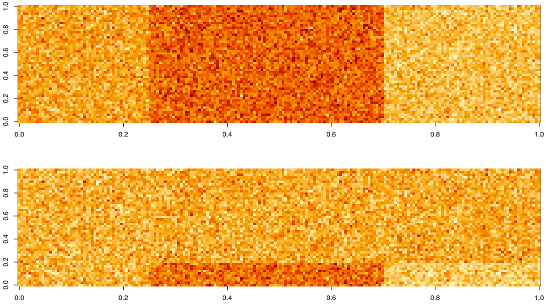


Figure 1: Visualisation of the data matrix of $n = 50$, $T = 200$ with two change points at $\eta_1 = 50$ and $\eta_2 = 150$, where the changes are dense (top) and sparse (bottom).

Aggregating methods in the literature

Some existing works consider different ways of aggregating the CUSUM series,

$$\mathcal{C}^i_{p,q,r} = \sqrt{\frac{r-q}{(r-p+1)(q-p+1)}} \sum_{t=p}^{q} X_{i,t} - \sqrt{\frac{q-p+1}{(r-p+1)(r-q)}} \sum_{t=q+1}^{r} X_{i,t}. \tag{3}$$

1. ℓ_2-aggregation (Horváth and Hušková, 2012; Zhang et al., 2010)

$$\max_q \left[\frac{q(T-q)}{\sqrt{n}T^2} \sum_{i=1}^{n} \left\{ (\mathcal{C}^i_{1,q,T})^2 - 1 \right\} \right], \tag{4}$$

2. ℓ_∞-aggregation (Jirak, 2015)

$$\max_q \max_i \left(\frac{q(T-q)}{T} \right)^{1/2} |\mathcal{C}^i_{1,q,T}|. \tag{5}$$

3. ℓ_1-aggregation of the hard-thresholded CUSUM (Cho and Fryzlewicz, 2015)

$$\max_q \sum_{i=1}^{n} |\mathcal{C}^i_{1,q,T}| \cdot \mathbb{I}\{|\mathcal{C}^i_{1,q,T}| > \lambda\}, \tag{6}$$

Aggregating methods in the literature

- ℓ_2-aggregation of the CUSUM series is good for detecting dense (but gentle) changes
- ℓ_∞-aggregation is more useful for capturing sparse (but large) changes.

Q. What if both sparse and dense changes exist in the multiple change scenario?

$\rightarrow$ We can use both ℓ_2 and ℓ_∞ aggregations!

Table of Contents

Introduction

Problem Setup

Our Bottom-up Algorithm
- Wavelet transform
- Thresholding
- Inverse wavelet transform

Theoretical results

Simulation results

Data applications

Summary

Outline of the algorithm

1. Wavelet transformation
   Perform a bottom-up unbalanced wavelet transform of the data matrix by recursively applying the conditionally orthonormal transformations to the same local regions of all vectors $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n$ in a bottom-up way.
2. Thresholding
   Perform a hard-thresholding set to zero those detail coefficients whose (aggregated) magnitude is smaller than a pre-specified threshold.
3. Inverse wavelet transformation
   Carry out the inverse transformation with the thresholded coefficients in step 2 which gives the initial estimates of $\boldsymbol{f}_1, \ldots, \boldsymbol{f}_n$ that is ℓ_2-consistent.

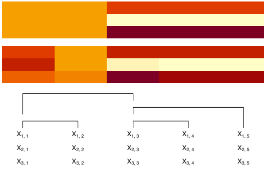


Figure 2: Wavelet transform

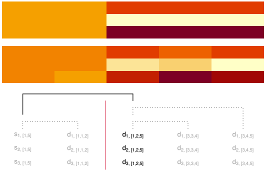


Figure 3: Thresholding

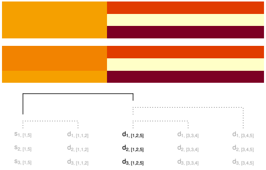


Figure 4: Inverse transform

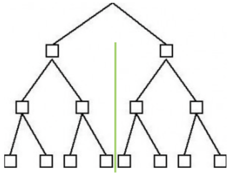
1. Wavelet transform

- The bottom-up data adaptive wavelet transformation produces a data-adaptive multiscale decomposition of the data matrix with detail and smooth coefficients.

$$\begin{pmatrix} s_{1,1} & d_{1,1} & \dots & d_{1,T-1} \\ s_{2,1} & d_{2,1} & \dots & d_{2,T-1} \\ \vdots & \vdots & \vdots & \vdots \\ s_{n,1} & d_{n,1} & \dots & d_{n,T-1} \end{pmatrix}_{n\times T} = \begin{pmatrix} X_{1,1} & \dots & X_{1,T} \\ X_{2,1} & \dots & X_{2,T} \\ \vdots & \vdots & \vdots \\ X_{n,1} & \dots & X_{n,T} \end{pmatrix}_{n\times T} \begin{pmatrix} \psi_{(0,1)} & \psi_1 & \dots & \psi_{T-1} \end{pmatrix}_{T\times T}$$

- This transform is performed in a way to push the ℓ_2 energy of the input data to a small number of detail coefficients.

- The bulk of variability (= deviation from constancy) of the signal tends to be mainly captured by few detail-type coefficients arising at the later stages of the transform. This sparse representation of the data justifies thresholding in the next step.



Example of the wavelet transformation

- Consider the initial input data matrix of the dimension 3×5,

$$\boldsymbol{S}_0 = \begin{pmatrix} X_{1,1} & X_{1,2} & X_{1,3} & X_{1,4} & X_{1,5} \\ X_{2,1} & X_{2,2} & X_{2,3} & X_{2,4} & X_{2,5} \\ X_{3,1} & X_{3,2} & X_{3,3} & X_{3,4} & X_{3,5} \end{pmatrix}$$

- We compare 4 pairs of columns by computing the detail vector for each pair of columns as follows:

$$\begin{bmatrix} X_{1,1} & X_{1,2} \\ X_{2,1} & X_{2,2} \\ X_{3,1} & X_{3,2} \end{bmatrix} \to \begin{bmatrix} d_{1,[1,1,2]} \\ d_{2,[1,1,2]} \\ d_{3,[1,1,2]} \end{bmatrix} \to \begin{pmatrix} d^{\ell 2}_{[1,1,2]} \\ d^{\ell \infty}_{[1,1,2]} \end{pmatrix}, \begin{bmatrix} X_{1,2} & X_{1,3} \\ X_{2,2} & X_{2,3} \\ X_{3,2} & X_{3,3} \end{bmatrix} \to \begin{bmatrix} d_{1,[2,2,3]} \\ d_{2,[2,2,3]} \\ d_{3,[2,2,3]} \end{bmatrix} \to \begin{pmatrix} d^{\ell 2}_{[2,2,3]} \\ d^{\ell \infty}_{[2,2,3]} \end{pmatrix},$$

$$\begin{bmatrix} X_{1,3} & X_{1,4} \\ X_{2,3} & X_{2,4} \\ X_{3,3} & X_{3,4} \end{bmatrix} \to \begin{bmatrix} d_{1,[3,3,4]} \\ d_{2,[3,3,4]} \\ d_{3,[3,3,4]} \end{bmatrix} \to \begin{pmatrix} d^{\ell 2}_{[3,3,4]} \\ d^{\ell \infty}_{[3,3,4]} \end{pmatrix}, \begin{bmatrix} X_{1,4} & X_{1,5} \\ X_{2,4} & X_{2,5} \\ X_{3,4} & X_{3,5} \end{bmatrix} \to \begin{bmatrix} d_{1,[4,4,5]} \\ d_{2,[4,4,5]} \\ d_{3,[4,4,5]} \end{bmatrix} \to \begin{pmatrix} d^{\ell 2}_{[4,4,5]} \\ d^{\ell \infty}_{[4,4,5]} \end{pmatrix}.$$

where $d_{i,[p,q,r]}$ is the detail coefficient obtained from merging $\{X_{i,p}, \ldots, X_{i,q}\}$ and $\{X_{i,q+1}, \ldots, X_{i,r}\}$.

- We use the aggregated details coefficient $d_{[p,q,r]}$.

How to make the algorithm robust? (Maeng et al., 2022)

- We can combine information of several ℓ_p norms using the rank of each series of detail coefficients.

$$
\begin{aligned}
(d^{\ell_2}_{[1,1,2]}, d^{\ell_2}_{[2,2,3]}, d^{\ell_2}_{[3,3,4]}, d^{\ell_2}_{[4,4,5]}) &\quad\rightarrow\quad (r^{\ell_2}_{[1,1,2]}, r^{\ell_2}_{[2,2,3]}, r^{\ell_2}_{[3,3,4]}, r^{\ell_2}_{[4,4,5]}) \\
(d^{\ell_\infty}_{[1,1,2]}, d^{\ell_\infty}_{[2,2,3]}, d^{\ell_\infty}_{[3,3,4]}, d^{\ell_\infty}_{[4,4,5]}) &\quad\rightarrow\quad (r^{\ell_\infty}_{[1,1,2]}, r^{\ell_\infty}_{[2,2,3]}, r^{\ell_\infty}_{[3,3,4]}, r^{\ell_\infty}_{[4,4,5]})
\end{aligned}
$$

  where r is the rank e.g. rank $= 1$ is the smallest.

- We work with the combined rank and merge the pair whose combined rank

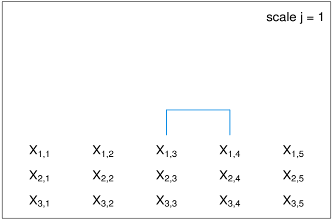
$$
\begin{pmatrix} r^*_{[1,1,2]} \\ r^*_{[2,2,3]} \\ r^*_{[3,3,4]} \\ r^*_{[4,4,5]} \end{pmatrix} = \begin{pmatrix} \max(r^{\ell_2}_{[1,1,2]}, r^{\ell_\infty}_{[1,1,2]}) \\ \max(r^{\ell_2}_{[2,2,3]}, r^{\ell_\infty}_{[2,2,3]}) \\ \max(r^{\ell_2}_{[3,3,4]}, r^{\ell_\infty}_{[3,3,4]}) \\ \max(r^{\ell_2}_{[4,4,5]}, r^{\ell_\infty}_{[4,4,5]}) \end{pmatrix}
$$

  is the smallest.

- If either $r^{\ell_2}_{\cdot}$ or $r^{\ell_\infty}_{\cdot}$ is very large then the corresponding pair is less likely to be merged in that pass.

Example of the Wavelet transformation (cont)

- Suppose $r^*_{[3,3,4]}$ has the smallest rank and merge the corresponding pair:



- Then update the initial input matrix into:

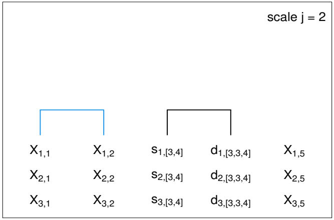
$$\boldsymbol{S} = \begin{pmatrix} X_{1,1} & X_{1,2} & s_{1,[3,4]} & d_{1,[3,3,4]} & X_{1,5} \\ X_{2,1} & X_{2,2} & s_{2,[3,4]} & d_{2,[3,3,4]} & X_{2,5} \\ X_{3,1} & X_{3,2} & s_{3,[3,4]} & d_{3,[3,3,4]} & X_{3,5} \end{pmatrix}.$$

- From now on, we ignore any detail coefficient columns in the updated data matrix. The possible pairs of neighbouring columns for the next merging are:

$$\begin{bmatrix} X_{1,1} & X_{1,2} \\ X_{2,1} & X_{1,2} \\ X_{3,1} & X_{1,2} \end{bmatrix}, \begin{bmatrix} X_{1,2} & s_{1,[3,4]} \\ X_{2,2} & s_{2,[3,4]} \\ X_{3,2} & s_{3,[3,4]} \end{bmatrix}, \begin{bmatrix} s_{1,[3,4]} & X_{1,5} \\ s_{2,[3,4]} & X_{2,5} \\ s_{3,[3,4]} & X_{3,5} \end{bmatrix}.$$

Example of the Wavelet transformation (cont)

- Assume that $r^*_{[1,1,2]}$ is the smallest rank among $r^*_{[1,1,2]}$, $r^*_{[2,2,4]}$, $r^*_{[3,4,5]}$, then we merge that pair:



- Then the data matrix is now updated into:

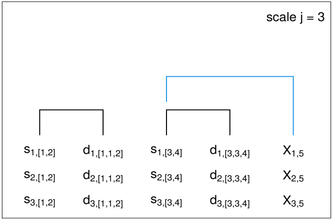
$$\boldsymbol{S} = \begin{pmatrix} s_{1,[1,2]} & d_{1,[1,1,2]} & s_{1,[3,4]} & d_{1,[3,3,4]} & X_{1,5} \\ s_{2,[1,2]} & d_{2,[1,1,2]} & s_{2,[3,4]} & d_{2,[3,3,4]} & X_{2,5} \\ s_{3,[1,2]} & d_{3,[1,1,2]} & s_{3,[3,4]} & d_{3,[3,3,4]} & X_{3,5} \end{pmatrix}. \tag{7}$$

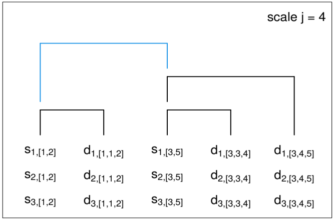
- We now compare the following two candidates for merging,

$$\begin{bmatrix} s_{1,[1,2]} & s_{1,[3,4]} \\ s_{2,[1,2]} & s_{2,[3,4]} \\ s_{3,[1,2]} & s_{3,[3,4]} \end{bmatrix}, \begin{bmatrix} s_{1,[3,4]} & X_{1,5} \\ s_{2,[3,4]} & X_{2,5} \\ s_{3,[3,4]} & X_{3,5} \end{bmatrix}.$$

Example of the Wavelet transformation (cont)

- Continue to merge as long as at least one pair of columns is available to be merged:





Summary of the Wavelet transformation

- At each scale j, for each pair of neighbouring smooth coefficients, $(s_{i,[p,q]}, s_{i,[q+1,r]})$, compute the corresponding $d_{i,[p,q,r]}$ for $i = 1, \dots, n$

$$\begin{aligned} d_{i,[p,q,r]} &= \sqrt{\frac{r-q}{r-p+1}} s_{i,[p,q]} - \sqrt{\frac{q-p+1}{r-p+1}} s_{i,[q+1,r]}, \\ &= \sqrt{\frac{r-q}{(r-p+1)(q-p+1)}} \sum_{t=p}^{q} X_{i,t} - \sqrt{\frac{q-p+1}{(r-p+1)(r-q)}} \sum_{t=q+1}^{r} X_{i,t}, \\ &= \langle \boldsymbol{X}_{i,[p:r]}, \psi_{p,q,r} \rangle, \end{aligned}$$

where $s_{i,[p,r]} = (r-p+1)^{-1/2} \sum_{t=p}^{r} X_{i,t}$.

- The corresponding wavelet function $\psi_{p,q,r}$ has a form of piecewise-constant function.
- Merging a pair of smooth coefficients is done through the orthonormal transform:

$$\begin{pmatrix} s_{i,[p,r]} \\ d_{i,[p,q,r]} \end{pmatrix} = \begin{pmatrix} -b_{p,q,r} & a_{p,q,r} \\ a_{p,q,r} & b_{p,q,r} \end{pmatrix} \begin{pmatrix} s_{i,[p,q]} \\ s_{i,[q+1,r]} \end{pmatrix}, \quad i = 1, \dots, n.$$

Summary of the Wavelet transformation

- $d_{i,[p,q,r]}$ computed from a subregion $\{X_{i,p}, \ldots, X_{i,r}\}$ of i^{th} data sequence represents the strength of the corresponding local polynomial trend.
- The smaller the (absolute) size of the detail, the smaller the local deviation from constancy in piecewise-constant scenario (and from linearity in piecewise-linear scenario).
- Using both ℓ_2 and ℓ_∞ aggregations makes the algorithm robust in sparseness of the change by delaying the merge of regions in which either dense (and gentle) or sparse (and large) change exists.

Pros and cons of bottom-up algorithm

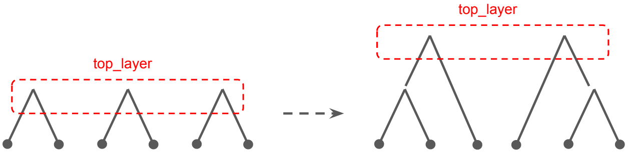
- Outperforms when there exist frequent change-points (with short segments).
- Detects both long and short segments well.

- However, it tends to give larger localisation error than the existing competitors.
- This is because the initial merges are done with too short segments thus affected a lot by outliers.

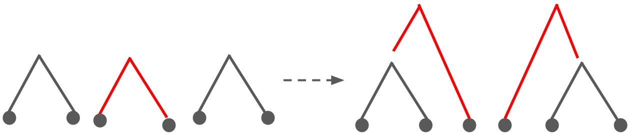
Q. Can we add a layer of "unmerge" to improve localisation error?

Localisation error improvement strategy - Data structure

- Binary tree structure: collection of nodes.
- Each node records its left and right children, as well as the length and mean value of the segment it captures.
- A vector recording the topmost layer of nodes.

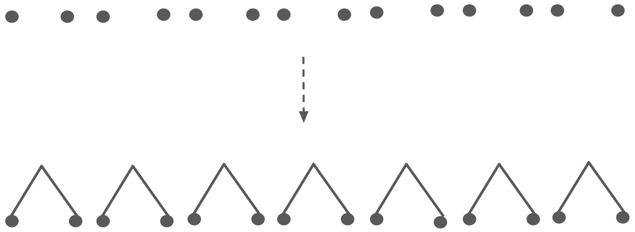


Localisation error improvement strategy - (1) Unmerge



- Allow unmerge if the left and right subsegments in a segment are respectively more similar to the left and right neighbouring segments.
- So each 'unmerge' is in fact one unmerge and two merges and the total number of segments still goes down by 1, otherwise, it becomes an infinite loop and does not terminate.
- Can optionally allow leftmost and rightmost segment to unmerge and group with their neighbour.

Localisation error improvement strategy - (2) Pre-merge



- Merge all top-layer nodes in pairs. When number of top-layer nodes is even, do (1,2), (3,4), . . . , (m-1, m). Otherwise, preferentially merge the extreme top-layer node with a shorter associated segment.
- Not to have segments of length one that includes an outlier.
- To have more reliable CUSUM statistics computed from segments whose length is not too short.

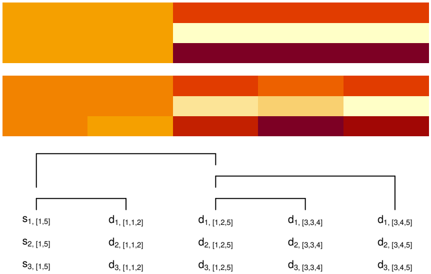
Localisation error improvement strategy - algorithm

1. Premerge + Unmerge: For the first few rounds, do one premerge pass, then run unmerges until no further change, then repeat.

2. Merge + Unmerge: do one merge pass, then run unmerges until no further change, then repeat.

1. Wavelet transform

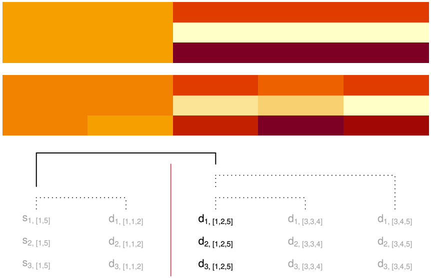
- The bottom-up data adaptive wavelet transformation produces a data-adaptive multiscale decomposition of the data matrix with detail and smooth coefficients.



- The data matrix of the dimension 3×5 is decomposed into one column of smooth coefficients and 4 columns of detail coefficients.

2. Thresholding

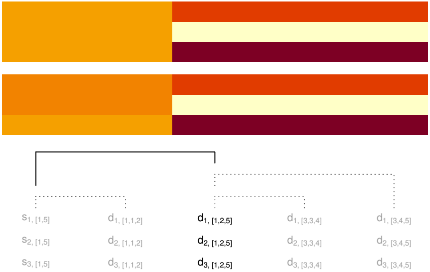
- Threshold away the details coefficients if both the ℓ_2 and the ℓ_∞ aggregations are less than the corresponding pre-specified threshold.



- Different thresholds are used for ℓ_2 and ℓ_∞ aggregations of the detail coefficients: $\lambda_{\ell_2} = C_1\sqrt{n}$ for ℓ_2 and $\lambda_{\ell_\infty} = C_2\sqrt{2\log nT}$ for ℓ_∞.

3. Inverse wavelet transform

- Using the thresholded detail coefficients, invert (= transpose) the local orthonormal transformations in reverse order to that in which they were originally performed.



- As the transform is orthonormal, the inverse transfrom can be done as

$$\begin{pmatrix} s_{i,[p,q]} \\ s_{i,[q+1,r]} \end{pmatrix} = \begin{pmatrix} -b_{p,q,r} & a_{p,q,r} \\ a_{p,q,r} & b_{p,q,r} \end{pmatrix}^{-1} \begin{pmatrix} s_{i,[p,r]} \\ d_{i,[p,q,r]} \end{pmatrix}, \quad i = 1, \ldots, n,$$

  where the thresholded detail coefficients are used.
- Through this inverse transform, the estimator $\{\tilde{\boldsymbol{f}}_i\}_{i=1}^n$ of the true signal $\{\boldsymbol{f}_i\}_{i=1}^n$ is obtained.

Table of Contents

Introduction

Problem Setup

Our Bottom-up Algorithm

- Wavelet transform
- Thresholding
- Inverse wavelet transform

Theoretical results

Simulation results

Data applications

Summary

Theoretical results

Recall the model:

$$X_{i,t} = f_{i,t} + \varepsilon_{i,t}, \quad i = 1, \dots, n,\ t = 1, \dots, T, \tag{8}$$

where $\boldsymbol{f}_i = (f_{i,1}, \dots, f_{i,T})^\top$ is the underlying signal vector of the observation $\boldsymbol{X}_i = (X_{i,1}, \dots, X_{i,T})^\top$. For each i, $\boldsymbol{\varepsilon}_i = (\varepsilon_{i,1}, \dots, \varepsilon_{i,T})^\top$ is the independent Gaussian random error with the conditions that $E(\boldsymbol{\varepsilon}_i) = 0$, $\text{Var}(\boldsymbol{\varepsilon}_i) = \sigma_i^2 = 1$.

We assume that the signal vectors $\{\boldsymbol{f}_i\}_{i=1}^n$ share N distinct change-points,

$$0 = \eta_0 < \eta_1 < \eta_2 < \dots < \eta_N < \eta_{N+1} = T, \tag{9}$$

in that at each change-point η_ℓ, there exists at least one signal $\boldsymbol{f}_i$ in which the trend change occurs at f_{i,η_ℓ}. Between any adjacent change-points, η_ℓ and $\eta_{\ell+1}$, the signals $\{\boldsymbol{f}_i\}_{i=1}^n$ have a form of piecewise-constant:

$f_{i,t} = \theta_{i,\ell}$ for $t \in [\eta_{\ell-1} + 1, \eta_\ell],\ \ell = 1, \dots, N+1,$ where $\exists\, \Omega_\ell = \left\{i : \left|f_{i,\eta_\ell+1} - f_{i,\eta_\ell}\right| \neq 0\right\} \subset \left\{1, \dots, n\right\}$ s.t. $\Omega_\ell \neq \emptyset$ for $\ell = 1, \dots, N$.

Theoretical results

Theorem

$\{\boldsymbol{X}_i\}_{i=1}^n$ follow model (8) and ($\{\hat{\boldsymbol{f}}_i\}_{i=1}^n$, $\hat{N}$) are the estimators obtained from the HiTs algorithm. Let Assumptions 1-3 hold and
$\lambda = C_1\{2\log(nT)\}^{1/2}$ with a constant $C_1 \geq \log\left(\frac{n^2T^4}{\sqrt{\log(nT)}}\right)/\log(nT)$
and the dimensionality n satisfies $n \sim T^\alpha$ for some fixed $\alpha \in (0,\infty)$.
Suppose that the number of true change-points, N, has the order of $\log T$. Assume that $nTR_{n,T} = o\Bigg(\min_\ell \left\{\left(\frac{1}{\Delta_{n,T}^\ell} + \frac{1}{\Delta_{n,T}^{\ell+1}}\right)^{-1} \cdot \delta_{n,T}^\ell\right\}\Bigg)$
where $\Delta_{n,T}^\ell = \sum_{i\in\Omega_\ell}\left(f_{i,\eta_\ell+1} - f_{i,\eta_\ell}\right)^2$, $\delta_{n,T}^\ell = |\eta_{\ell+1} - \eta_\ell|$,
$\|\tilde{\boldsymbol{f}} - \boldsymbol{f}\|_{n,T}^2 = O_p(R_{n,T})$, $R_{n,T} = N(\max_\ell \mathcal{S}_\ell)(nT)^{-1}\log(T)\log(nT)$
and $\mathcal{S}_\ell$ is the cardinality of the set Ω_ℓ.
Then we have

$$\mathbb{P}\left(\hat{N} = N, \quad \max_{\ell=1,\dots,N}\left\{|\hat{\eta}_\ell - \eta_\ell| \cdot \Delta_{n,T}^\ell\right\} \leq CnTR_{n,T}\right) \to 1, \quad (10)$$

as $n, T \to \infty$ where C is a constant.

Table of Contents

Introduction

Problem Setup

Our Bottom-up Algorithm
- Wavelet transform
- Thresholding
- Inverse wavelet transform

Theoretical results

Simulation results

Data applications

Summary

Comparing the results with competitors

1. **HiTS**: High-dimensional Trend Segmentation via bottom-up algorithms (our proposal)
2. **INSPECT**: Informative Sparse projection (Wang and Samworth (2018))
3. **SBS**: Sparsified Binary Segmentation (Cho and Fryzlewicz (2015))
4. **DC**: Double Cusum (Cho (2016))

Two change-points

# of time points	T = 300
# of data sequences	n = 500
change-point locations	$\eta_1 = 100$, $\eta_2 = 200$
sparsity	sparse: $\mathcal{S}_1 = 25$, $\mathcal{S}_2 = 25$
	dense: $\mathcal{S}_1 = 350$, $\mathcal{S}_2 = 350$
	mixed: $\mathcal{S}_1 = 350$, $\mathcal{S}_2 = 25$
change magnitude	$\theta_1' = 3.1/\sqrt{\mathcal{S}_1}$
	$\theta_2' = 3.1/\sqrt{\mathcal{S}_2}$

Table 1: Simulation setting, where θ_j is root mean squared change magnitude in coordinates that undergo the j-th change for each changepoint.

Simulation results - sparse

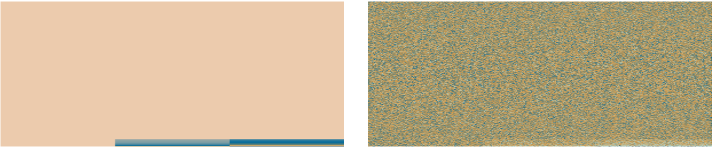


Figure 5: True signal matrix (left) and the data with added noise (right) when both change-points are sparse.

Method	$\hat{N} - N$							$d_H(\times 10^2)$
	$\leq$-3	-2	-1	0	1	2	$\geq$3	
HITS	0	0	0	**100**	0	0	0	1.83
INSPECT	0	0	0	**78**	18	2	2	3.18
SBS	0	0	0	**93**	7	0	0	2.27
DC	0	0	0	**52**	37	8	3	5.73

Table 2: Distribution of $\hat{N} - N$ and the average Hausdorff distance d_H over 100 simulation runs.

Simulation results - dense

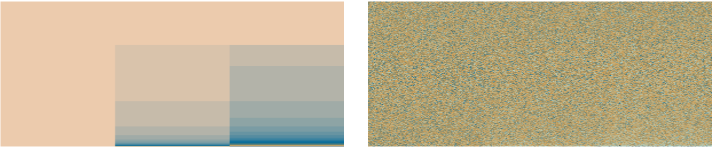


Figure 6: True signal matrix (left) and the data with added noise (right) when both change-points are dense.

Method	$\hat{N} - N$							$d_H(\times 10^2)$
	$\leq$-3	-2	-1	0	1	2	$\geq$3	
HITS	0	0	2	**98**	0	0	0	3.14
INSPECT	0	0	0	**79**	16	3	2	3.39
SBS	0	0	0	**91**	9	0	0	3.17
DC	0	0	0	**92**	8	0	0	2.84

Table 3: Distribution of $\hat{N} - N$ and the average Hausdorff distance d_H over 100 simulation runs.

Simulation results - mixed

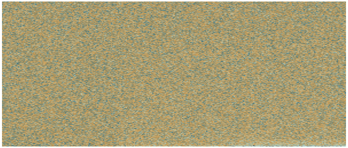
 

Figure 7: True signal matrix (left) and the data with added noise (right) when the signal has both sparse and dense changes.

Method	$\hat{N}-N$							$d_H(\times 10^2)$
	≤-3	-2	-1	0	1	2	≥3	
HITS	0	0	0	**100**	0	0	0	2.04
INSPECT	0	0	0	**77**	19	2	2	3.23
SBS	0	0	12	**86**	2	0	0	5.65
DC	0	0	0	**70**	25	4	1	3.94

Table 4: Distribution of $\hat{N}-N$ and the average Hausdorff distance d_H over 100 simulation runs.

Table of Contents

Introduction

Problem Setup

Our Bottom-up Algorithm
- Wavelet transform
- Thresholding
- Inverse wavelet transform

Theoretical results

Simulation results

Data applications

Summary

Data example: January temperatures in South Africa

We analyse January average temperature in 50 cities of South Africa from 1857 to 2013.

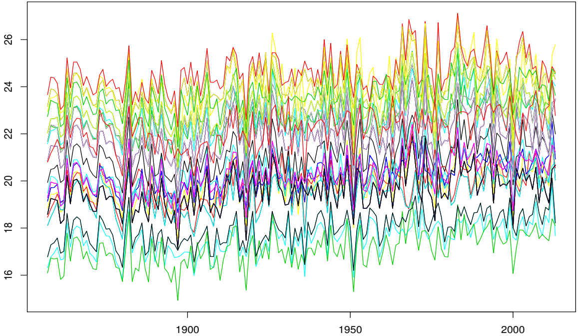


Figure 8: January average temperature curves of 50 cities in South Africa from 1857 to 2013.

Data example (continued)

The HiTS algorithm identifies 2 change-points, 1912 and 1965.

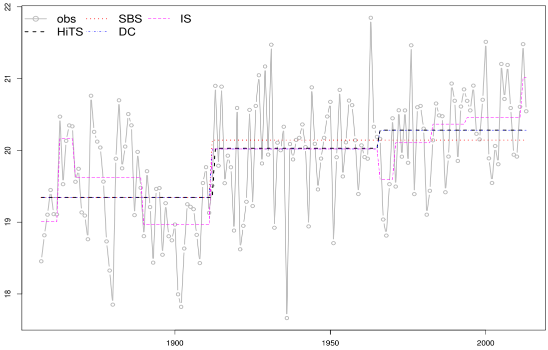


Figure 9: The data series (grey dots) of Cape town and estimated signal with change-points.

Table of Contents

Introduction

Problem Setup

Our Bottom-up Algorithm
- Wavelet transform
- Thresholding
- Inverse wavelet transform

Theoretical results

Simulation results

Data applications

Summary

Summary and comments

- We propose a robust bottom-up algorithm for detecting multiple change-points in mean of the high dimensional panel data where the dimension n can increase polynomially fast with respect to the length of the series T.
- The bottom-up nature of the algorithm work well in detecting both long and short subsegments.
- Using both ℓ_2 and ℓ_∞ aggregation enable the algorithm to detect both sparse and dense changes in which other existing methods fail.
- Our algorithm reduces the computational complexity by merging multiple regions in a single pass over the data.
- The consistency in estimating the number and the locations of change-points is shown in the i.i.d. Gaussian noise setting and can be extended to the temporal dependence.
- The algorithm can be extended to the piecewise-linear scenario.

References I

Cho, H. (2016). Change-point detection in panel data via double cusum statistic. *Electronic Journal of Statistics*, 10:2000–2038.

Cho, H. and Fryzlewicz, P. (2015). Multiple-change-point detection for high dimensional time series via sparsified binary segmentation. *Journal of the Royal Statistical Society Series B*, 77:475–507.

Fryzlewicz, P. (2018). Tail-greedy bottom-up data decompositions and fast mulitple change-point detection. *Ann. Stat.*, 46:3390–3421.

Horváth, L. and Hušková, M. (2012). Change-point detection in panel data. *Journal of Time Series Analysis*, 33:631–648.

Jirak, M. (2015). Uniform change point tests in high dimension. *Ann. Stat.*, 43:2451–2483.

Maeng, H., Wang, T., and Fryzlewicz, P. (2022). Multiple trend-change detection for high-dimensional time series. *In preparation*.

Vostrikova, L. (1981). Detection of the disorder in multidimensional random-processes. *Doklady Akademii Nauk SSSR*, 259:270–274.

Wang, T. and Samworth, R. J. (2018). High dimensional change point estimation via sparse projection. *Journal of the Royal Statistical Society Series B*, 80:57–83.

Zhang, N. R., Siegmund, D. O., Ji, H., and Li, J. Z. (2010). Detecting simultaneous changepoints in multiple sequences. *Biometrika*, 97:631–645.