Filtered derivative with p-value method for multiple change points detection

Pierre R. BERTRAND[1,2], Mehdi FHIMA[2] and Arnaud GUILLIN[2]

[1] INRIA Saclay, France

[2] Laboratoire de Mathématiques, CNRS UMR 6620 & University Clermont-Ferrand II, France

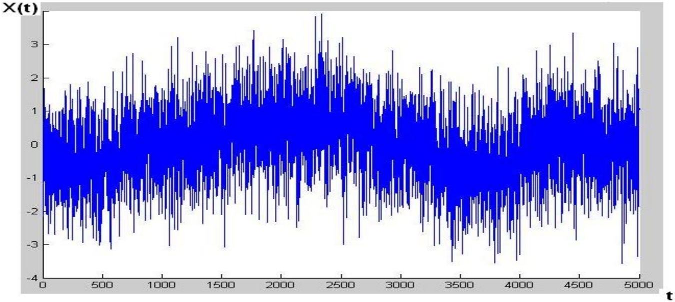
Neuvième Colloque "Jeunes Probabilistes et Statisticiens", 3rd May 2010

Plan

1. **Presentation of the problem**
   - Our problem
   - Penalized method
   - Filtered Derivative with p-value
2. **Description of the FDp-V method**
   - Filtered Derivative method
   - Filtered Derivative with p-value method
   - Complexity
3. **Numerical results**
   - Monte Carlo simulation
   - Detection of change points in the slope
   - Segmentation of the electrocardiogram (ECG)

Our model

Let $(X_1, \ldots, X_n)$ be a sequence of independent r.v.'s presenting change points in a parameter $\theta = (\theta_1, \ldots, \theta_n)$.

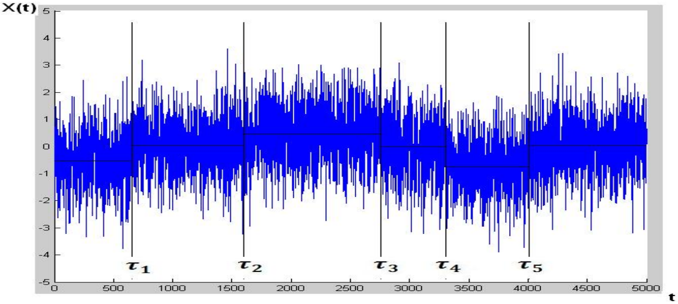


Toy model: Simulated data of independent Gaussian r.v.'s presenting change points in the mean.

Our goal

Estimate:

- The change points $\tau = (\tau_1, \ldots, \tau_K)$,
- The parameter $\hat{\theta}_k$ of each segment $(\tau_{k-1} + 1, \tau_k)$.



$\rightarrow$ **Off-line detection problem.**

Penalized least square criterion (PLSC) method

1. If the number of change points is **known**
    - Bai & Perron (1994).
    - Lavielle & Moulines (2000).
2. If the number of change points is **unknown**
    - Birgé & Massart (2006).
    - Lavielle & Teyssière (2006).

Presentation of the method

- The contrast function can be written as follows:

$$J(K,\mathbf{T},X) = \frac{1}{n}\sum_{k=1}^{K+1} U\left(X_{T_{k-1}+1},\ldots,X_{T_k}\right)$$

  where $U\left(X_{T_{k-1}+1},\ldots,X_{T_k}\right)$ is the contrast function for estimating θ in the segment $(T_{k-1}+1, T_k)$.

- $(\tau_1,\ldots,\tau_K) = \arg\min_{\mathbf{T}} J(K,\mathbf{T},X)$.

- The dynamic programming algorithm is a method which essentially proceeds via a sequential examination of the overall $U\left(X_{i+1},\ldots,X_j\right)$.

Presentation of the method

- The **penalized** contrast function can be written as follows:

$$J(K,\mathbf{T},X) = \frac{1}{n}\sum_{k=1}^{K+1} U\left(X_{T_{k-1}+1},\dots,X_{T_k}\right) + \beta \mathbf{pen}(\mathbf{T})$$

  where $U\left(X_{T_{k-1}+1},\dots,X_{T_k}\right)$ is the contrast function for estimating θ in the segment $(T_{k-1}+1, T_k)$.

- $(\tau_1,\dots,\tau_K) = \arg\min_{\mathbf{T}} J(K,\mathbf{T},X)$.

- The dynamic programming algorithm is a method which essentially proceeds via a sequential examination of the overall $U\left(X_{i+1},\dots,X_j\right)$.

Complexity

This algorithm needs to compute and store the upper triangular matrix:

$$M = \left(U\left(X_{i+1},\ldots,X_j\right)\right)_{1\leq i\leq j\leq n}$$

Time complexity

Computation of the values $U\left(X_{i+1},\ldots,X_j\right)$ needs time complexity of $\mathcal{O}(n^2)$.

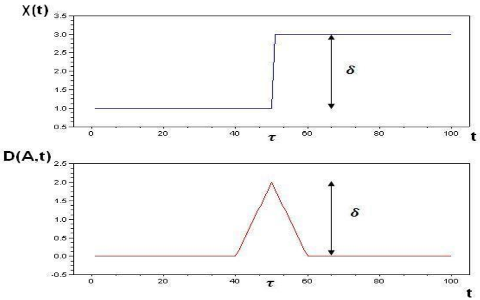
Memory complexity

Storage of a matrix of size n.
$\Rightarrow$ Memory allocation: $8n^2$ bytes.

Filtered Derivative with p-value (FDp-V) method

1. **Filtered Derivative method**
   - Basseville & Nikiforov (1993).
   - Antoch & Huskova (1994).
   - Bertrand (2000).
2. **Filtered Derivative with p-value method**
   - Bertrand & Fhima (2009)
   - Bertrand & Fhima & Guillin (2010)

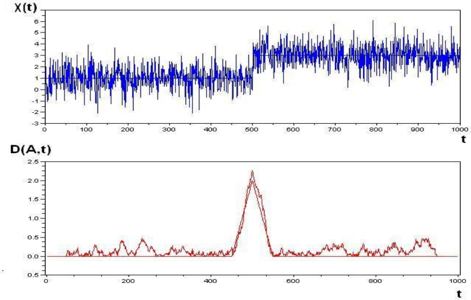
Initial idea: "Hat function"



Where

$$D(k,A) = \begin{cases} \hat{\theta}(k+1, k+A) - \hat{\theta}(k-A+1, k) & \text{if} \quad k \in [A, n-A] \\ 0 & \text{else} \end{cases}$$

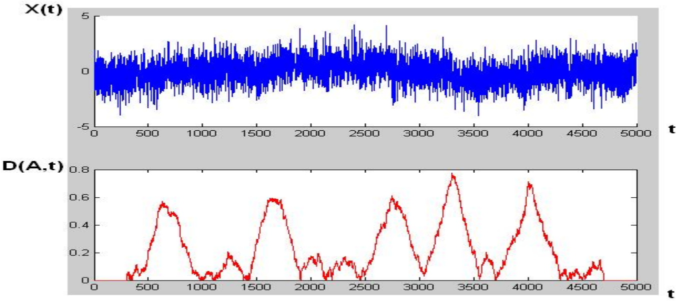
Real data



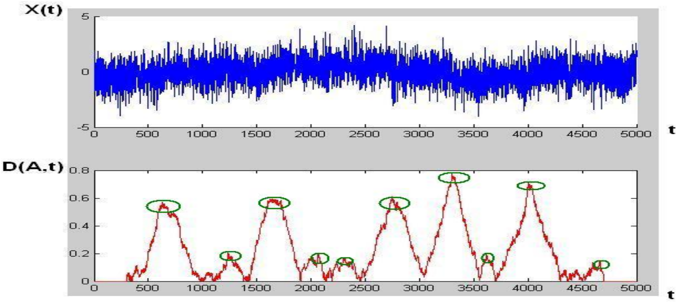
Where

$$D(k,A) = \begin{cases} \hat{\theta}(k+1, k+A) - \hat{\theta}(k-A+1, k) & \textit{if} \quad k \in [A, n-A] \\ 0 & \textit{else} \end{cases}$$

Real data with multiple change-points

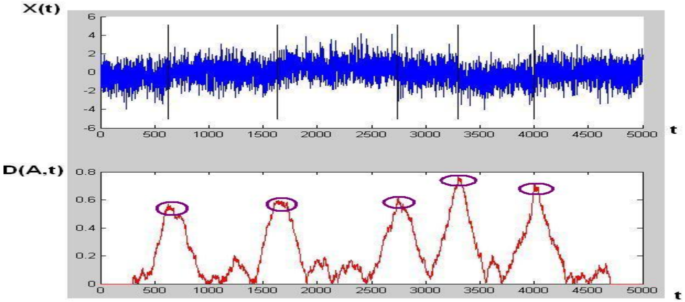


Real data with multiple change-points



1. How do we proceed to **detect potential change points** $(\widetilde{\tau}_1, \ldots, \widetilde{\tau}_{K_{\max}})$? → Step 1.

Real data with multiple change-points



1. How do we proceed to **detect potential change points** $(\widetilde{\tau}_1, \ldots, \widetilde{\tau}_{K_{\max}})$? $\rightarrow$ Step 1.
2. How do we **eliminate false alarms** and keep only right change points that we denote $(\widehat{\tau}_1, \ldots, \widehat{\tau}_K)$? $\rightarrow$ Step 2.

Step 1: Detection of the potential change points

It is based on the following test

$$(H_0):\ \theta_1 = \theta_2 = \cdots = \theta_{n-1} = \theta_n$$

against

$(H_1):\ \exists K \geq 1$ and $0 = \tau_0 < \tau_1 < \cdots < \tau_K < \tau_{K+1} = n$ such that

$$\theta_1 = \cdots = \theta_{\tau_1} \neq \theta_{\tau_1+1} = \cdots = \theta_{\tau_2} \cdots \neq \theta_{\tau_K+1} = \cdots = \theta_{\tau_{K+1}}.$$

Step 1: Detection of the potential change points

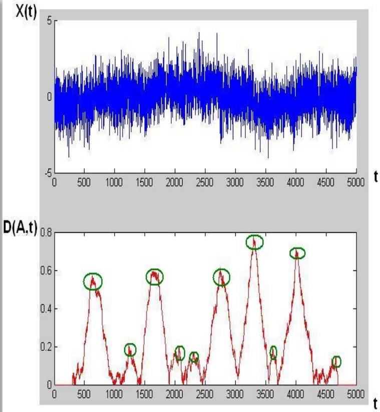
Step 1 of the algorithm

1. Fix type I error at level p_1^* and so the critical value C_1 given by

$$P\left(\max_{k\in[A:n-A]} |D(k,A)| > C_1\right) = p_1^*(C_1).$$

2. Select as potential change points, $\widetilde{\tau}_k$, local maxima for which $|D(\widetilde{\tau}_k, A)| > C_1$.

$\Rightarrow$ We obtain potential change points $(\widetilde{\tau}_1, \dots, \widetilde{\tau}_{Kmax})$.

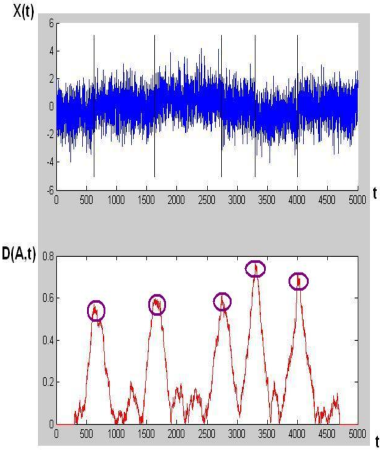


Step 2: False alarms elimination

Step 2 of the algorithm

1. For each $\widetilde{\tau}_k$, associate the p-value $\widetilde{p}_k$.
2. Keep only the change points $\widetilde{\tau}_k$ such that

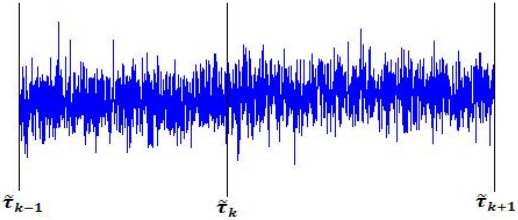
$$\widetilde{p}_k < p_2^*$$



Statistical hypothesis testing

Let

- $(H0_k) : \widehat{\theta}_k = \widehat{\theta}_{k+1}$
- $(H1_k) : \widehat{\theta}_k \neq \widehat{\theta}_{k+1}$



We compute p-values $(\widetilde{p}_1, \ldots, \widetilde{p}_{K_{\max}})$ associated respectively with each potential change points $(\widetilde{\tau}_1, \ldots, \widetilde{\tau}_{K_{\max}})$.

Complexity

Time complexity

To calculate the "hat function" we use recurrence formula, i.e. that

$$D(k+1, A) = f\left(D(k, A)\right)$$

For instance, in the case of change points in the mean, we have the following recurrence formula

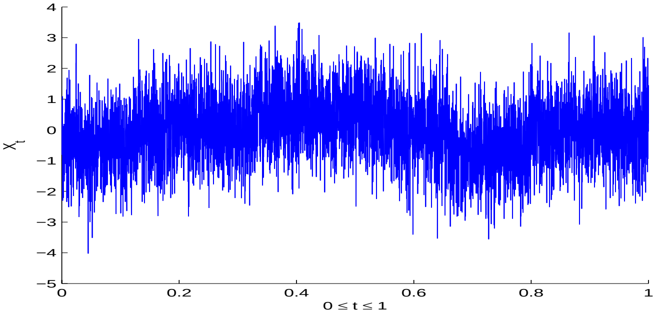
$$D(k+1, A) = D(k, A) + A^{-1}\left[X_{k+A+1} - 2X_k + X_{k-A+1}\right]$$

$\Rightarrow$ Complexity of $\mathcal{O}(n)$.

Memory complexity

Storage of a vector of size n.
$\Rightarrow$ Memory allocation: $8n$ bytes.

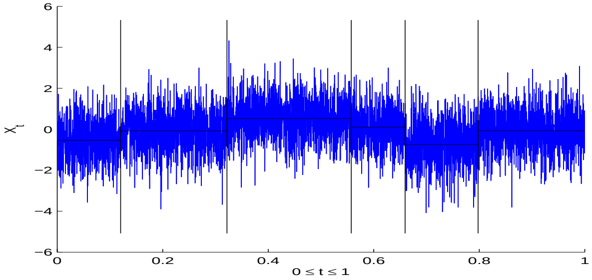
Signal to be segmented



Data:

- Independent Gaussian r.v,
- $n = 5000$,
- $\tau = \{0.1294, 0.3232, 0.5532, 0.66, 0.8\}$,
- $\delta_k \in [0.5, 1.25]$.

Calibration of the algorithms



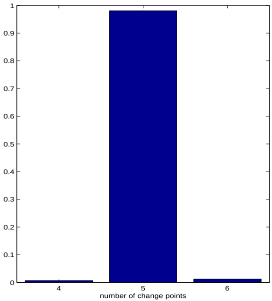
PLSC method

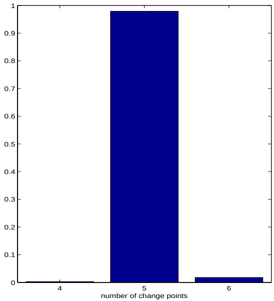
1. $K_{max} = 10$
2. p-value$= 10^{-5}$

FDp-V method

1. $p_1^* = 0.05$
2. $p_2^* = 10^{-5}$
3. $A = 300$

Number of change points

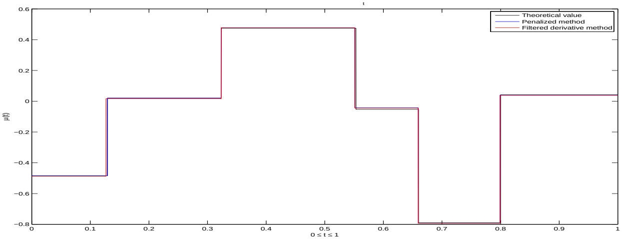




Results:

- PLSC (left) : $K = 5$ in 97.9% of all cases.
- FDp-V (right): $K = 5$ in 98.1% of all cases.

Estimation errors



	SECP	MISE
FDp-V	1.1840×10^{-4}	0.0107
PLSC	1.2947×10^{-4}	0.0114

Where:

- Square Error on Change Points (SECP)= $\mathbb{E}\|\widehat{g} - g\|^2_{L^2(0,1)}$
- Mean Integrated Squared Error (MISE)= $\mathbb{E}\|\widehat{\tau} - \tau\|^2$

Complexity

	Memory allocation	CPU time
FDp-V method	0.04 MB	0.005 s
PLSC method	200 MB	240 s

Conclusion

FDp-V method:

- Faster (time)
- Cheaper (memory)

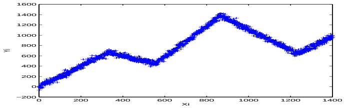
Detection of change points in the slope

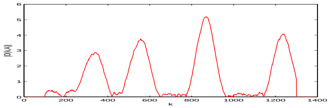
Data

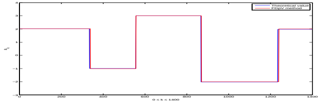
1. $n = 1400$
2. $\Delta = 1$
3. $\sigma = 30$
4. $\nu_k \in [3, 5]$ where $\nu_k := |a_k - a_{k+1}|$.

Calibration of the FDp-V method

1. $p_1^* = 0.05$
2. $p_2^* = 10^{-5}$
3. $A = 100$







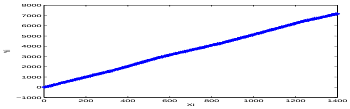
Smaller change points

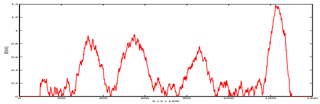
Data

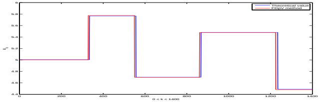
1. $n = 1400$
2. $\Delta = 1$
3. $\sigma = 30$
4. $\nu_k \in [0.75, 1]$.

Calibration of the FDp-V method

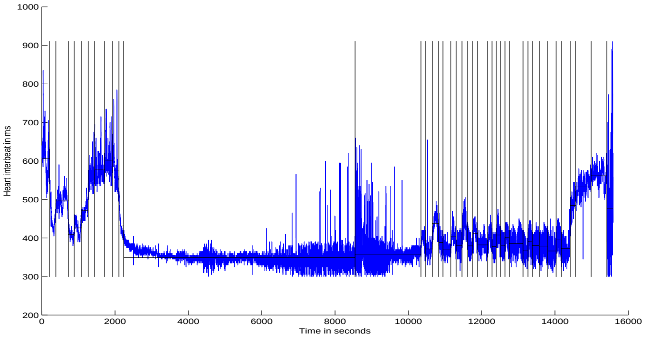
1. $p_1^* = 0.05$
2. $p_2^* = 10^{-5}$
3. $A = 100$







Segmentation of the electrocardiogram (ECG)



Segmentation of heartbeat time series of a marathon runner.