Détection d'une rupture offline et online

Session : Détection de ruptures

Journée MAS 2022

G. RIGAILL

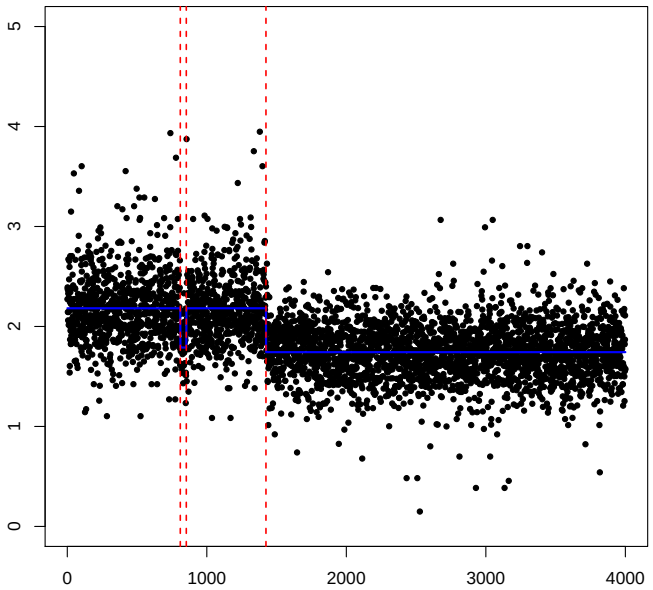
IPS2 (Gnet) et LaMME (Stat & Genome)

30 Août 2022

Plan

Un exemple de données

Analyse du nombre de copies d'ADN [Pierre-Jean *et al.* 2015]



- Détecter des ruptures dans la moyenne
- Plusieurs objectifs possibles
  1. Estimer le nombre de ruptures
  2. Estimer la position des ruptures
  3. Résumer l'information

Modèle constant par morceaux

Avec plusieurs ruptures

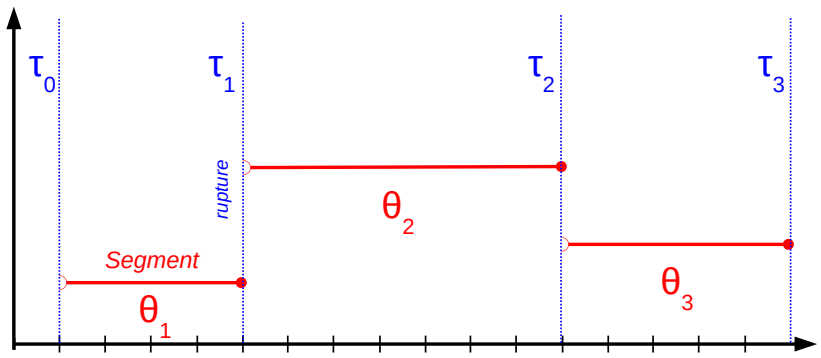
données $Y_1, \ldots, Y_n$

ruptures $\tau = (\tau_1, .., \tau_D)$

segments $s_d = (\tau_{d-1}, \tau_d]$

paramètres $\theta = (\theta_1, .., \theta_D)$

modèle $Y_i \sim \mathcal{F}(\theta_d)$ i.i.d



Possibles objectifs, estimation des paramètres

- « Détecter » les ruptures ($\approx$ estimer le nombre de ruptures)
- « Localiser » les ruptures
- « Débruiter » le signal

Difficulté statistique & algorithmique

- $\binom{n}{2}$ segments et 2^{n-1} segmentations

Une littérature abondante

- De nombreuses applications : biologie, finance . . .
- Explosion du nombre de méthodes ces dernières années
  - Pour des modèles de plus en plus complexes
  - Mais aussi pour le modèle Gaussien univarié i.i.d [Harchaoui and Levy-Leduc 2009, Killick *et al.* 2011, Frick *et al.* 2014, Dette and Wied 2015, Maidstone *et al.* 2017, Fryzlewicz 2017 . . .]
- Très grande diversité d'approches [Truong *et al.* 2018, Niu et Zhang 2016, . . .]

Une idée récurrente

- Se ramener à la détection d'une rupture dans un « intervalle »

Détection d'une rupture

Pourquoi ?

- Un problème plus simple mais encore étudié
  - Même dans le cas Gaussien

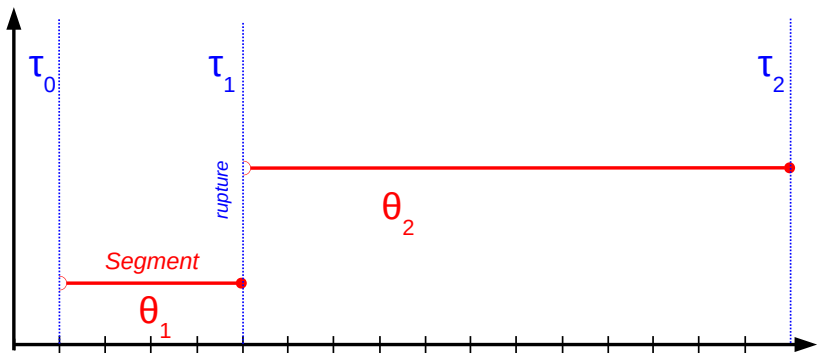
    [Verzelen *et al.* 2020, Maillard 2019, Yu *et al.* 2020, Romano *et al.* 2022]
- Le ressort de nombreuses méthodes et preuves
  - Méthodes à base de segmentation binaire [Scott and Knott 1974, Fryzlewicz 2014, Anastasiou et Fryzlewicz 2019 . . .]
  - Cas particulier de programmation dynamique [Bellman 1962, Auger et Lawrence 1989 . . .]
  - Preuves de garanties statistiques . . . [Zheng *et al.* 2019, Pilliat *et al.* 2021 . . .]
  - . . .

Plan

Modèle constant par morceaux

Avec une rupture

données	$Y_1, \ldots, Y_n$
rupture	τ_1
segments	$(0, \tau_1], (\tau_1, n]$
paramètres	$\theta_1 \neq \theta_2$?
modèle	$Y_i \sim \mathcal{F}(\theta_d)$ i.i.d



Difficulté statistique & algorithmique ?

Inférence avec une rupture

Rapport de vraisemblance et statistique Cumsum

- Modèle : $Y_i \sim \mathcal{N}(\theta_d, \sigma^2)$, avec $\sigma^2 = 1$
- Y a-t-il une rupture ?
- Comparer le modèle avec ou sans la rupture τ

$$
\begin{aligned}
LR_\tau &= \sum_{i=1}^{\tau}(y_i - \bar{y}_{1:\tau})^2 + \sum_{i=\tau+1}^{n}(y_i - \bar{y}_{\tau+1:n})^2 - \sum_{i=1}^{n}(y_i - \bar{y}_{1:n})^2 \\
C_\tau &= \sqrt{\frac{\tau(n-\tau)}{n}}\,|\bar{y}_{1:\tau} - \bar{y}_{\tau+1:n}| \qquad \text{[Cumsum]}
\end{aligned}
$$

Avec une rupture

Objectif

- On va chercher la meilleure rupture τ

$$\arg \min_{1 \leqslant \tau \leqslant n} LR_\tau \qquad \text{ou} \qquad \arg \max_{1 \leqslant \tau \leqslant n} C_\tau$$

- Stratégie
  - Algorithmiquement : calculer le minimum/maximum
  - Statistiquement : contrôler les variations

Plan

Programmation dynamique sur la dernière rupture

Algorithme v1 - « offline » pour une rupture dans $Y_{1:n}$

« Segment Neighborhood » pour une rupture : $\mathcal{O}(n)$

[Scott et Knott 1974, Auger et Lawrence 1989] [a]

$$\arg\min_{\tau} LR_{\tau} = \arg\min_{1 \leqslant \tau < n} \left\{ \underbrace{\sum_{i=1}^{\tau} (y_i - \bar{y}_{1:\tau})^2}_{c_{1:\tau}} + \underbrace{\sum_{i=\tau+1}^{n} (y_i - \bar{y}_{\tau+1:n})^2}_{c_{\tau+1:n}} \right\}$$

a. $c_{1:n}^{(D+1)} = \min_{\tau} \{ c_{1:\tau}^{(D)} + c_{\tau+1:n}^{(1)} \}$

Calcul en $\mathcal{O}(1)$ des $c_{i:j}$

- Pré-calcul de $S_{1:j} = \sum_{i=1}^{j} y_i$

Programmation dynamique sur la dernière rupture

Algorithme v1 pour plusieurs ruptures

- Utilisé dans la Segmentation binaire et dérivés

Amélioration ?

- Peu efficace dans un cadre « online » : $\mathcal{O}(n^2)$
- Idée, la solution pour $Y_{1:n}$ est proche de celle pour $Y_{1:n+1}$?
  - Programmation dynamique → élagage ?

Plan

Contrôle des variations de C_τ

En l'absence de rupture

$$C_\tau = \sqrt{\frac{\tau(n-\tau)}{n}} \, |\bar{y}_{1:\tau} - \bar{y}_{\tau+1:n}|$$

- Alors $\sqrt{\frac{\tau(n-\tau)}{n}} \, (\bar{y}_{1:\tau} - \bar{y}_{\tau+1:n}) \sim \mathcal{N}(0,1)$
- Donc $P(C_\tau \geqslant b) \leqslant 2e^{-\frac{b^2}{2}}$

$$P(C_\tau \geqslant b) \leqslant 2ne^{-\frac{b^2}{2}}$$

Contrôle des variations de C_τ

- Une pénalité par segment de $b^2 = 2\beta \log(n) + cste$

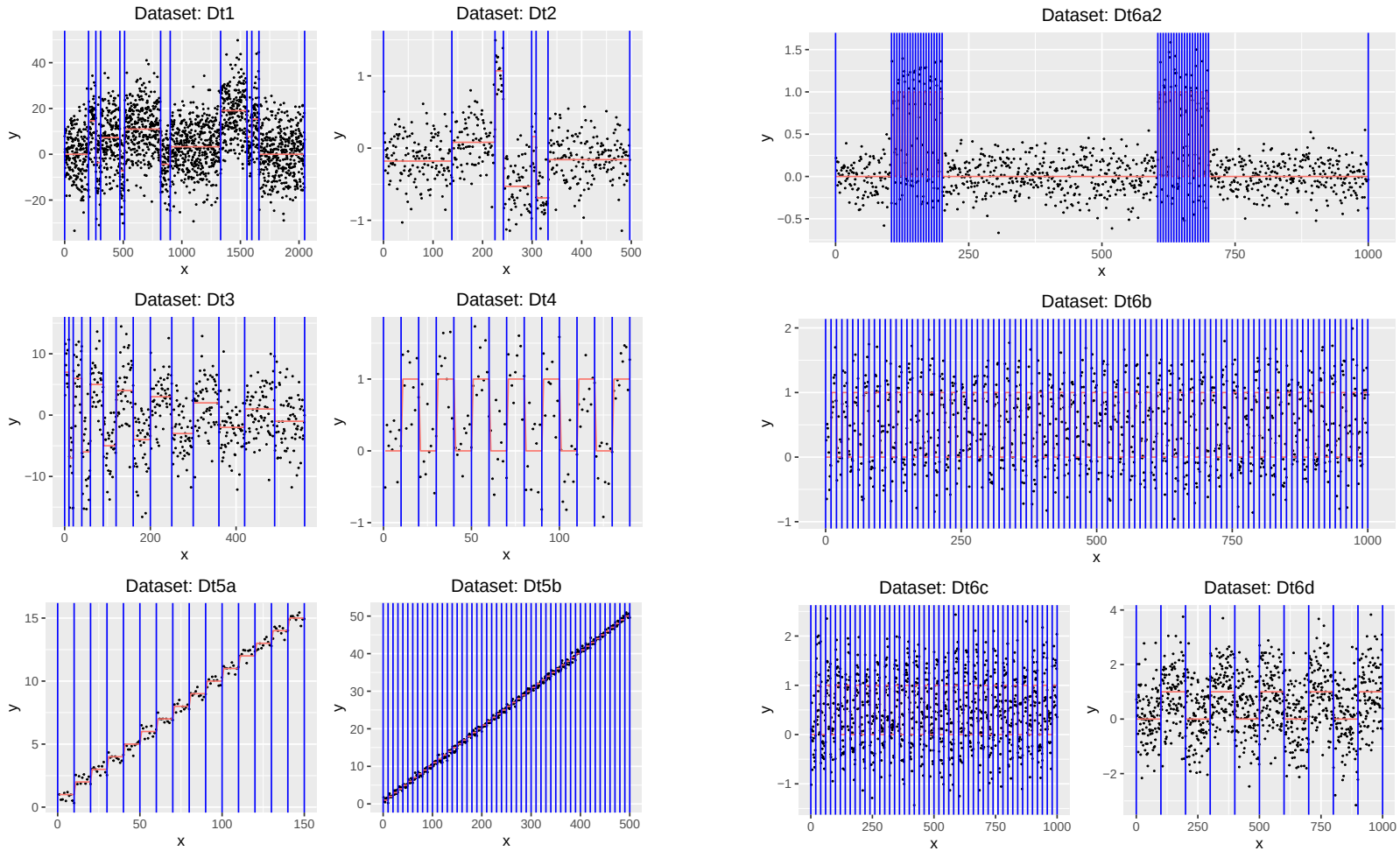
Pour plusieurs ruptures

- Cela fonctionne aussi : une borne d'union sur tous les segments

  [Fryzlewicz 2014]
- Consistant pour une asymptotique « infill » [Yao 1989 . . .]
- En pratique relativement efficace (assez facile à calibrer)

Quelques simulations [Fearnhead et Rigaill (2020)]

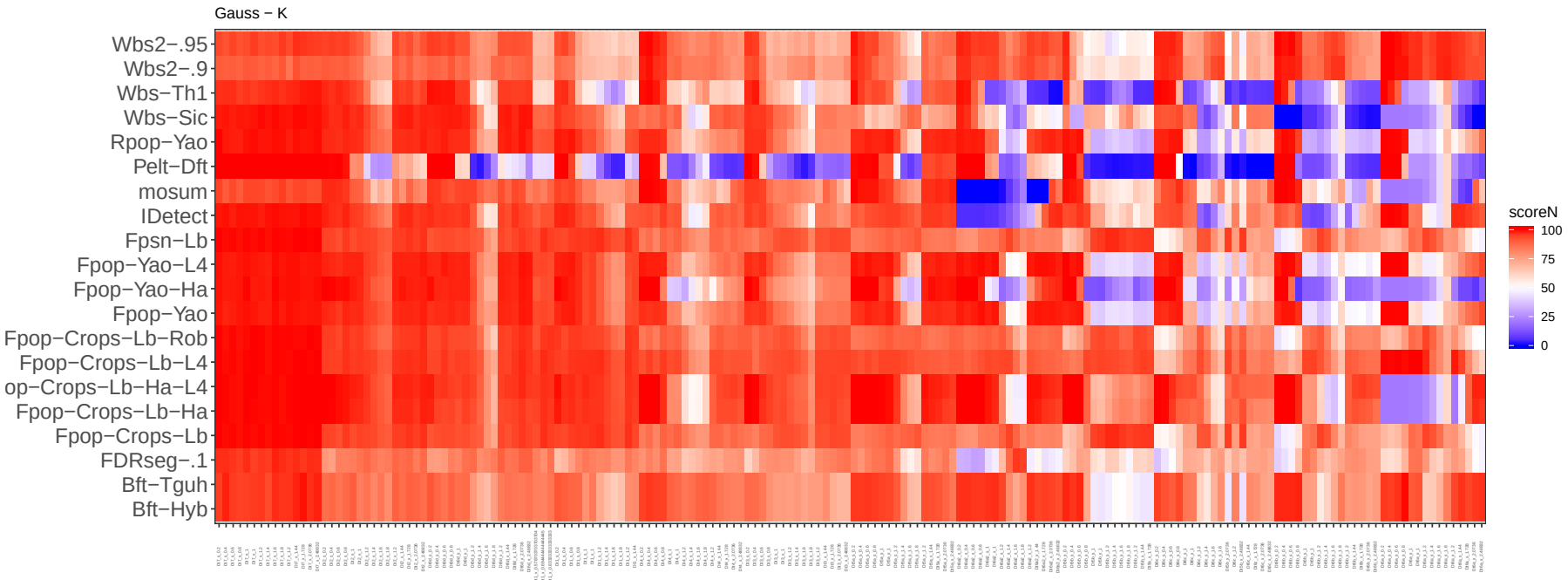
Faire varier n et σ^2 pour chaque type



Quelques simulations [Fearnhead et Rigaill (2020)]

Pourcentage de victoires/ex-aequos

- Pour chaque méthode M1
  - Pour chaque M2 compter le nombre de victoires/ex-aequos de M1
  - Sommer sur toutes les paires et ramener à 100



Pour aller plus loin

De meilleures garanties pour une rupture

- « Offline » - Multi-échelle [Verzelen *et al.* 2020]
- « Online » - [Maillard 2019, Yu *et al.* 2020]

De meilleures garanties pour plusieurs ruptures

- Contrôle du MSE : sélection de modèle [Lebarbier 2005, Arlot *et al.* 2016]
- Détection et localisation des ruptures : pénalités multi-échelles

  [Frick *et al.* 2013, Cho et Kirch 2020, Verzelen *et al.* 2020]
  - Pénaliser d'avantage les petits segments

Un peu d'épluchage statistique : « Peeling bound »

Qu'il faudra comparer à l'élagage algorithmique

Lemme 4 page 33 [Verzelen *et al.* 2020]

- $\varepsilon_1, \cdots, \varepsilon_n$ sous-gausiennes : $E[e^{s\varepsilon_i}] \leqslant e^{s^2/2}$ pour tout $s > 0$.
- Alors pour tout entier $d > 0$, $\alpha > 0$ et $x > 0$

$$P\left(\max_{k\in[d,d(1+\alpha)]} \frac{\sum_1^k \varepsilon_i}{\sqrt{k}} \geqslant x\right) \leqslant \exp\left(-\frac{x^2}{2(1+\alpha)}\right)$$

- Application avec $\alpha = 1$ et $d \in \{1, 2, 4, \cdots, 2^{\lfloor \log_2(n) \rfloor}\}$

Contrôle des variations de C_τ - 2

En l'absence de rupture - « Peeling bound »

$$C_\tau = \sqrt{\frac{\tau(n-\tau)}{n}} \, |\bar{y}_{1:\tau} - \bar{y}_{\tau+1:n}|$$

- Contrôler les $\frac{\sum_1^i y_t}{\sqrt{i}}$ et $\frac{\sum_{i+1}^n y_t}{\sqrt{n-i}}$
- Lemme 4 avec $\alpha = 1$ et $d \in \{1, 2, 4, \cdots, 2^{\lfloor \log_2(n) \rfloor}\}$
- En remarquant $(\sqrt{t} + \sqrt{n-t}) \leqslant \sqrt{2n}$

$$P(C_\tau \geqslant \sqrt{2}b) \leqslant 4 \log_2(n) e^{-\frac{b^2}{4}}$$

Plan

Programmation dynamique sur les moyennes

Algorithme v2 - « online » pour une rupture dans $Y_{1:\dots}$

- Idée : « Conditionner » [Johnson 2011-2013, Rigaill 2011-2015, Maidstone *et al.* 2017]

Vraisemblance sachant les moyennes θ_0 et θ_1

$$\mathcal{Q}_{\tau,n}(\theta_0,\theta_1) = -\sum_{t=1}^{\tau}(y_t-\theta_0)^2 - \sum_{t=\tau+1}^{n}(y_t-\theta_1)^2.$$

- Rapport de vraisemblance optimal :

$$LR_n = \max_{\substack{\tau\in\{1,\dots,n-1\}\\ \theta_0,\theta_1\in\mathbb{R}^2}} \{\mathcal{Q}_{\tau,n}(\theta_0,\theta_1)\} + \sum_{t=1}^{n}(y_t-\bar{y}_{1:n})^2$$

Programmation dynamique sur les moyennes

Comparaison de deux ruptures

- Comparer le coût de deux ruptures $i < j \leqslant n$

$$\mathcal{Q}_{i,n}(\theta_0, \theta_1) - \mathcal{Q}_{j,n}(\theta_0, \theta_1) = (\theta_1 - \theta_0) \left(2 \sum_{i+1}^{j} y_t - (j-i)(\theta_0 + \theta_1) \right)$$

- Changement de variables $\delta = \frac{1}{2}(\theta_1 - \theta_0)$ et $m = \frac{1}{2}(\theta_1 + \theta_0)$

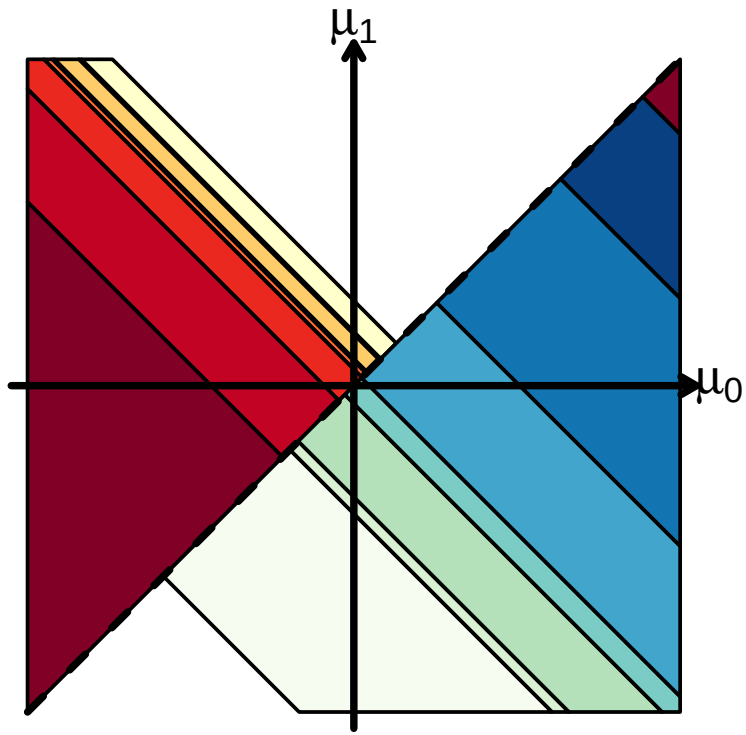
$$\mathcal{Q}_{i,n}(m, \delta) - \mathcal{Q}_{j,n}(m, \delta) = 4\delta(j-i) \left(\bar{y}_{i+1:j} - m \right)$$

Programmation dynamique sur les moyennes

Signe de $\mathcal{Q}_{i,n}(m,\delta) - \mathcal{Q}_{j,n}(m,\delta)$ pour $\delta > 0$

- Ne dépend pas de n
  - Positif pour $m \leqslant \bar{y}_{i+1:j}$
    i est meilleure
  - Négatif pour $m \geqslant \bar{y}_{i+1:j}$
    j est meilleure
- Zone de vie de i pour $\delta > 0$

$$m \in \left[\max_{0 \leqslant j < i} \bar{y}_{j+1:i} \ , \ \min_{i < j \leqslant n} \bar{y}_{i+1:j} \right]$$

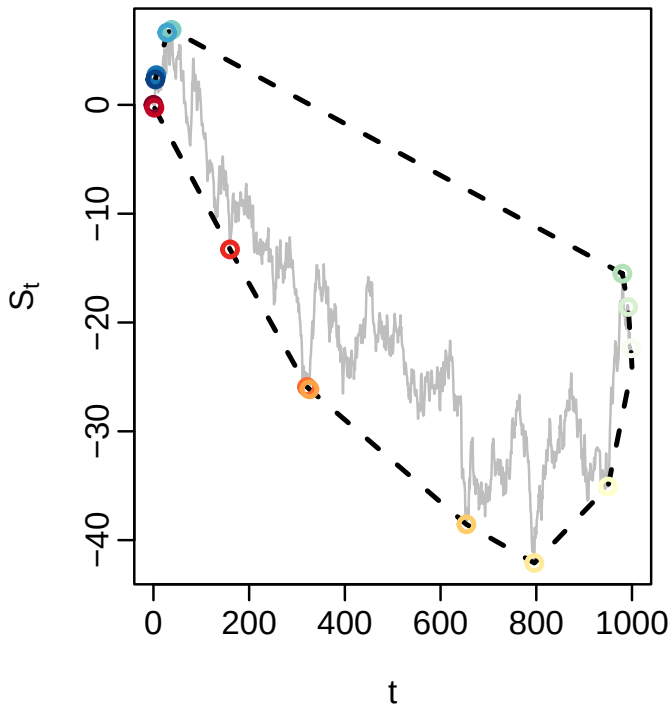


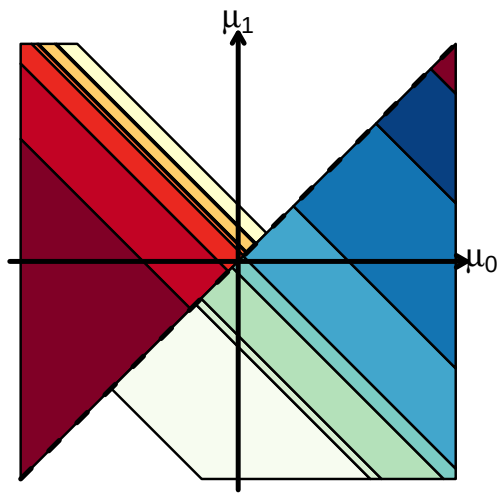
Programmation dynamique sur les moyennes

Enveloppe convexe de $S_{1:i} = \sum_{t=1}^{i} y_t$

- La rupture i n'est pas éliminée si

$$\forall\, j < i < j' \quad \frac{S_{j+1:i}}{i-j} < \frac{S_{i+1:j'}}{j'-i}$$





Recherche des points sur l'enveloppe convexe [Melkman 1989]

L'algorithme de Melkman est « online » et $\mathcal{O}(n)$

- Triplets ordonnés : rupture, $S_{1:\tau}$, borne de l'intervalle

$$Q = \{q_i = (\tau_i,\ s_i,\ l_i)\ \forall\ i = 1, \ldots,\ k\},$$

- Une étape (temps amorti de $\mathcal{O}(1)$

1. $q_{k+1} \leftarrow (\tau_{k+1} = n,\ s_{k+1} = S_n,\ l_{k+1} = \infty)$
2. $i \leftarrow k$
3. TANT QUE $(s_{k+1} - s_i - (\tau_{k+1} - \tau_i) l_i \leqslant 0$ et $i \geqslant 1)$ FAIRE $i - -$
4. $l_{k+1} \leftarrow (s_{k+1} - s_i)/(\tau_{k+1} - \tau_i)$
5. SI $i \neq k$ FAIRE $Q \leftarrow Q \backslash \{q_{i+1},\ \ldots, q_k\}$

Programmation dynamique sur les moyennes

Algorithme v2 - « online » pour une rupture dans $Y_{1:\dots}$

Pour chaque t

1. Mise à jour des points de l'enveloppe : Melkman
2. Maximisation du rapport de vraisemblance
   - en ne prenant en compte que les points sur l'enveloppe convexe

Nombre de points sur l'enveloppe convexe : $\#\mathcal{I}_{i:j}$

Etude pour une marché aléatoire des points sur l'enveloppe

[Andersen 1955]

Si les ε_t sont i.i.d continue sur $i : j$ alors

$$E(\#\mathcal{I}_{i:j}) \leqslant 2 \sum_{t=1}^{j-i-1} 1/(t+1) + 2 \leqslant 2\log(n) + 2$$

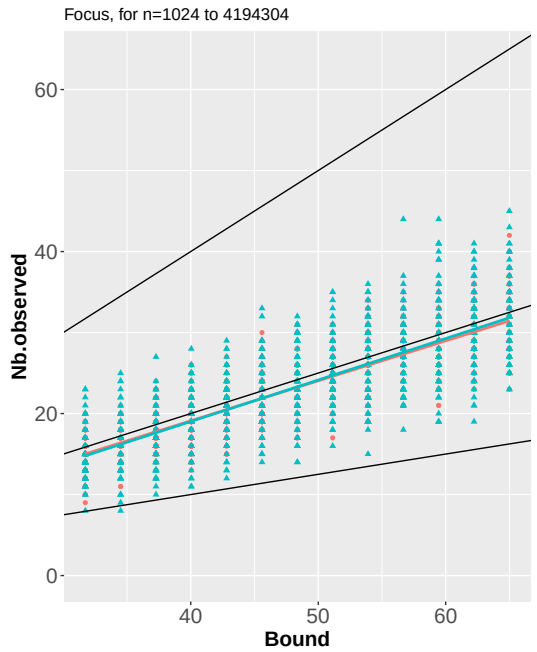
Algorithme v2 (FOCuS) est en espérance $\mathcal{O}(n\log(n))$

- Etape t en $\mathcal{O}(\log(t)) =$ *Melkman* : $\mathcal{O}(1) +$ *Andersen* : $\mathcal{O}(\log(t))$

Algorithme FOCuS en pratique

En l'absence de rupture

- Pour $n = 10^7$: 1 à 2 secondes



Segmentations maximisant la vraisemblance en 3 et 4 morceaux

- Utiliser FOCuS de 1 à n et de n à 1
- Temps de calcul en espérance de $\mathcal{O}(n\log(n))$

Plan

Conclusion, quelques perspectives

- Détection d'une rupture
  - Ce n'est pas si simple - « online »
  - Similarité élagage algorithmique et épluchage statistique : $\log(n)$
- Détection de plusieurs ruptures
  - Utile de bien comprendre le problème à une rupture
- Quelques perspectives
  - Programmation dynamique sur les moyennes et pénalités multi-échelle
  - Programmation dynamique sur les moyennes pour des modèles multivariés
  - Pour des structures de données plus complexes
  - Accélérer la BS et dérivés avec l'algorithme v2 « online »

Merci pour votre attention

- Gaetano Romano, Paul Fearnhead, Idris Eckley
- Relating and comparing methods for detecting changes in mean
  https://onlinelibrary.wiley.com/doi/full/10.1002/sta4.291
- Fast Online Changepoint Detection via Functional Pruning CUSUM statistics
  https://arxiv.org/abs/2110.08205