Change-point detection & clustering in a Poisson process

S. Robin

joint ongoing work with E. Lebarbier, C. Dion-Blanc

Sorbonne université

Stats au sommet, Rochebrune, Mar. 2024

Previously in Rochebrune

Bat cries

Previously in Rochebrune

Point process on $t \in [0, 1]$.

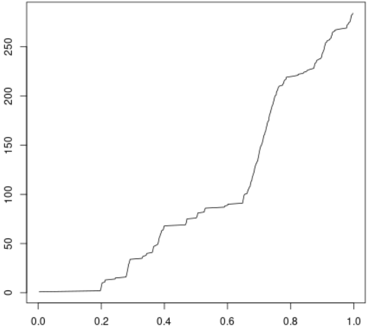
Event times:

$$0 < T_1 < \dots T_i < \dots T_n < 1$$

Counting process:

$$N(t) = \sum_{i=1}^{n} \mathbb{I}\{T_i \leqslant t\}$$

Bat cries[a]



[a] source: Vigie-Chiro program, Y. Bas, CESCO-MNHN

Previously in Rochebrune

Point process on $t \in [0, 1]$.

Event times:

$$0 < T_1 < \dots T_i < \dots T_n < 1$$

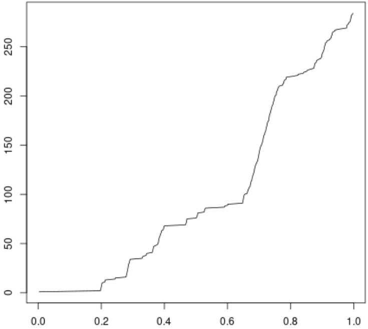
Counting process:

$$N(t) = \sum_{i=1}^{n} \mathbb{I}\{T_i \leqslant t\}$$

Poisson Process.

$$\{N(t)\}_{0 \leqslant t \leqslant 1} \sim PP(\lambda(t))$$

Bat cries[a]



[a]source: Vigie-Chiro program, Y. Bas, CESCO-MNHN

Intensity function $\lambda(t)$:

$$\lambda(t) = \lim_{\Delta t \to 0} \frac{\mathbb{P}\{N(t + \Delta t) - N(t) = 1\}}{\Delta t}, \qquad \mathbb{E}N(s) - \mathbb{E}N(t) = \int_t^s \lambda(u) \, \mathrm{d}u$$

Previously in Rochebrune

Piecewise constant intensity function.

Change-points

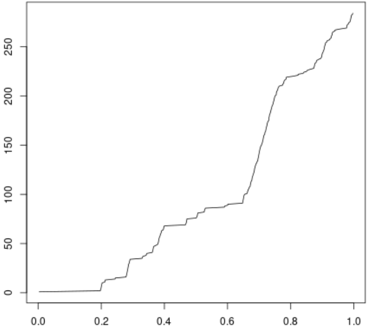
$$(\tau_0 =)0 < \tau_1 \cdots < \tau_{K-1} < 1 (= \tau_K)$$

For $t \in I_k =]\tau_{k-1}, \tau_k]$:

$$\lambda(t) = \lambda_k$$

$\rightarrow$ Continuous piece-wise linear cumulated intensity function

Bat cries[a]



[a] source: Vigie-Chiro program, Y. Bas, CESCO-MNHN

Previously in Rochebrune

Piecewise constant intensity function.

Change-points

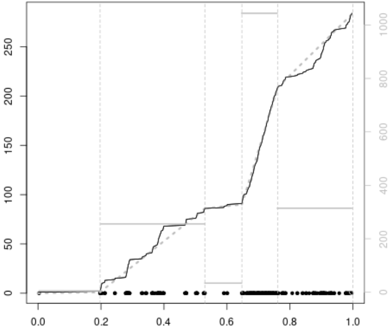
$$(\tau_0 =)0 < \tau_1 \cdots < \tau_{K-1} < 1(= \tau_K)$$

For $t \in I_k =]\tau_{k-1}, \tau_k]$:

$$\lambda(t) = \lambda_k$$

$\rightarrow$ Continuous piece-wise linear cumulated intensity function

Bat cries[a]



[a] source: Vigie-Chiro program, Y. Bas, CESCO-MNHN

Aim.

- Segmentation: estimate (τ, λ) reasonnably fast
- Model selection: choose K

Previously in Rochebrune

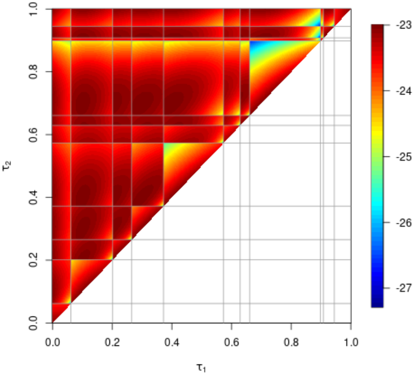
Efficient change-point detection. If the constrast (e.g. neg-log-likelihood) is

- additive wrt the segments and
- concave wrt the length of each segment,

Previously in Rochebrune

Efficient change-point detection. If the constrast (e.g. neg-log-likelihood) is

- additive wrt the segments and
- concave wrt the length of each segment,



Previously in Rochebrune

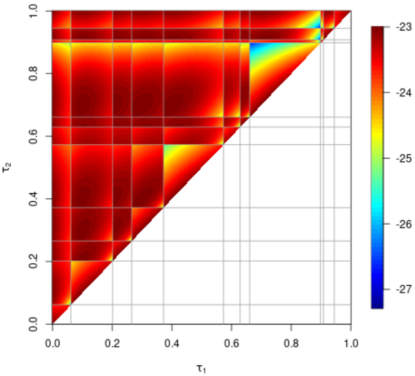
Efficient change-point detection. If the constrast (e.g. neg-log-likelihood) is

- additive wrt the segments and
- concave wrt the length of each segment,

Previously in Rochebrune

Efficient change-point detection. If the constrast (e.g. neg-log-likelihood) is

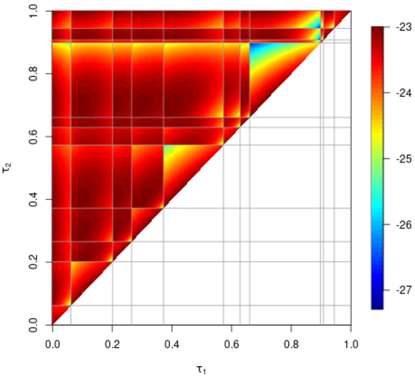
- additive wrt the segments and
- concave wrt the length of each segment,



Previously in Rochebrune

Efficient change-point detection. If the constrast (e.g. neg-log-likelihood) is

- additive wrt the segments and
- concave wrt the length of each segment,



then the optimal change points belong to

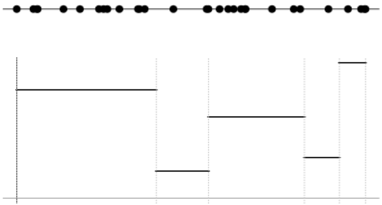
$$\{T_1^-, T_1, T_2^-, T_2, \dots T_i^-, T_i, \dots, T_n^-, T_n\}$$

→ Continuous to discrete optimization problem
→ Dynamic programming algorithm $= \mathcal{O}(n^2)$.

Previously in Rochebrune

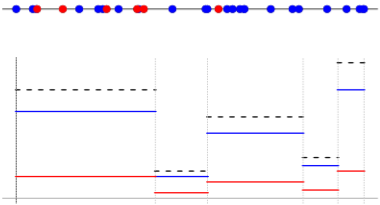
Lazy model selection. Thining property:

- independent processes with proportional internsities and common change point;
- cross-validation procedure to choose K.



[DBLR23]

Previously in Rochebrune



Lazy model selection. Thining property:

- independent processes with proportional internsities and common change point;
- cross-validation procedure to choose K.

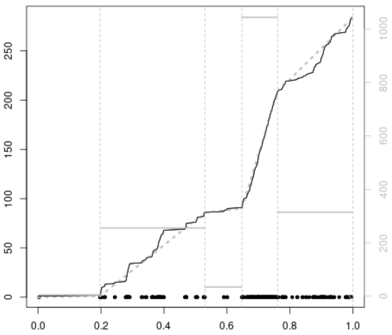
[DBLR23]

On-going. Consistency + (modified) BIC criterion for choosing K.

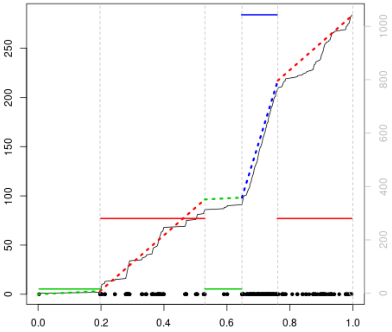
Next step: clustering

Next step: clustering

Next step: clustering



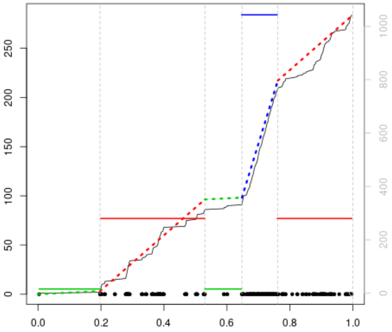
Next step: clustering



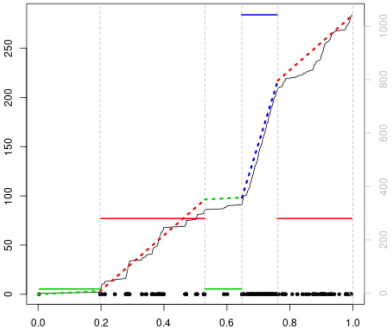
Next step: clustering

Segment clustering. P groups of segments

- each segments belong to one group
- group = underlying behavior



Next step: clustering



Segment clustering. P groups of segments

- each segments belong to one group
- group = underlying behavior

Model.

- K segments $I_1, \dots I_K$
- Segment k belongs to group p with probability π_p
- Point process $N(I_k) \sim PP(\lambda_p)$

Segmentation-clustering model

More precisely.

- $K + 1$ change-points: $\tau_0 = 0 < \tau_1 < \cdots < \tau_{K-1} < \tau_K = 1$
- K segments: $I_k = (\tau_{K-1}, \tau_K]$

Segmentation-clustering model

More precisely.

- $K+1$ change-points: $\tau_0 = 0 < \tau_1 < \cdots < \tau_{K-1} < \tau_K = 1$
- K segments: $I_k = (\tau_{K-1}, \tau_K]$
- P groups, with proportions: $\pi = (\pi_1, \dots \pi_K)$
- K latent variables (segment memberships):

$$(Z_k)_{1 \leqslant k \leqslant K} \text{ iid} \sim \mathcal{M}(1, \pi)$$

Segmentation-clustering model

More precisely.

- $K+1$ change-points: $\tau_0 = 0 < \tau_1 < \cdots < \tau_{K-1} < \tau_K = 1$
- K segments: $I_k = (\tau_{K-1}, \tau_K]$
- P groups, with proportions: $\pi = (\pi_1, \dots \pi_K)$
- K latent variables (segment memberships):

$$(Z_k)_{1 \leqslant k \leqslant K} \text{ iid} \sim \mathcal{M}(1, \pi)$$

- P intensities: $\lambda = (\lambda_1, \dots \lambda_K)$
- Point process in segment k:

$$N(Ik) \mid Z_k = p \sim PP(\lambda_p)$$

Segmentation-clustering model

More precisely.

- $K+1$ change-points: $\tau_0 = 0 < \tau_1 < \cdots < \tau_{K-1} < \tau_K = 1$
- K segments: $I_k = (\tau_{K-1}, \tau_K]$
- P groups, with proportions: $\pi = (\pi_1, \dots \pi_K)$
- K latent variables (segment memberships):

$$(Z_k)_{1 \leqslant k \leqslant K} \text{ iid} \sim \mathcal{M}(1, \pi)$$

- P intensities: $\lambda = (\lambda_1, \dots \lambda_K)$
- Point process in segment k:

$$N(Ik) \mid Z_k = p \sim PP(\lambda_p)$$

Model parameters.

$$\theta = (\pi, \lambda, \tau)$$

$+ (K, P)$

Inference algorithm (1/2)

Observed likelihood. Segment $I_k = (\tau_{K-1}, \tau_K]$, width $\Delta\tau_k$, count ΔN_k:

$$\log p_\theta(N) = \sum_{k=1}^{K} \log\left(\sum_{p=1}^{P} \pi_p p_{\lambda_p}(N(I_k))\right) = \sum_{k=1}^{K} \underbrace{\log\left(\sum_{p=1}^{P} \pi_p e^{-\lambda_p \Delta\tau_k} \lambda_p^{\Delta N_k}\right)}_{-c(I_k)}.$$

Inference algorithm (1/2)

Observed likelihood. Segment $I_k = (\tau_{K-1}, \tau_K]$, width $\Delta\tau_k$, count ΔN_k:

$$\log p_\theta(N) = \sum_{k=1}^K \log\left(\sum_{p=1}^P \pi_p p_{\lambda_p}(N(I_k))\right) = \sum_{k=1}^K \underbrace{\log\left(\sum_{p=1}^P \pi_p e^{-\lambda_p \Delta\tau_k} \lambda_p^{\Delta N_k}\right)}_{-c(I_k)}.$$

Proposition (concave constrast)

The contrast $c(I_k)$ is a concave function of the width $\Delta\tau_k = \tau_k - \tau_{k-1}$ of interval I_k

Inference algorithm (1/2)

Observed likelihood. Segment $I_k = (\tau_{K-1}, \tau_K]$, width $\Delta\tau_k$, count ΔN_k:

$$\log p_\theta(N) = \sum_{k=1}^{K} \log\left(\sum_{p=1}^{P} \pi_p p_{\lambda_p}(N(I_k))\right) = \sum_{k=1}^{K} \underbrace{\log\left(\sum_{p=1}^{P} \pi_p e^{-\lambda_p \Delta\tau_k} \lambda_p^{\Delta N_k}\right)}_{-c(I_k)}.$$

Proposition (concave constrast)

The contrast $c(I_k)$ is a concave function of the width $\Delta\tau_k = \tau_k - \tau_{k-1}$ of interval I_k

Consequence. For given mixture parameters (π, λ), the optimal segmentation

$$\widehat{\tau}(\pi, \lambda) = \arg\max_{\tau} \log p_{(\pi,\lambda,\tau)}(N)$$

is a subset of $\{T_1^-, T_1, T_2^-, T_2, \dots T_i^-, T_i, \dots, T_n^-, T_n\}$.

Inference algorithm (2/2)

Complete likelihood. Denoting $Z_{kp} = \mathbb{I}\{Z_k = p\}$,

$$\log p_\theta(N, Z) = \sum_{k=1}^{K} \sum_{p=1}^{P} Z_{kp} \left(\log \pi_p + \log p_{\lambda_p}(N(I_k)) \right)$$

Inference algorithm (2/2)

Complete likelihood. Denoting $Z_{kp} = \mathbb{I}\{Z_k = p\}$,

$$\log p_\theta(N, Z) = \sum_{k=1}^{K} \sum_{p=1}^{P} Z_{kp} \left(\log \pi_p + \log p_{\lambda_p}(N(I_k))\right)$$

'EM-DP' algorithm. (see [PRLD07] for a discrete time version)

Clustering. Given $\tau^{(h)}$, get

$$(\pi, \lambda)^{(h+1)} = \arg\max_{(\pi,\lambda)} \log p_{(\pi,\lambda,\tau^{(h)})}(N)$$

using expectation-maximization (EM), based on $\mathbb{E}_\theta(\log p_\theta(N, Z) \mid N)$;

Segmentation. Given $(\pi^{(h+1)}, \lambda^{(h+1)})$, get

$$\tau^{(h+1)} = \arg\max_{\tau} \log p_{(\pi^{(h+1)},\lambda^{(h+1)},\tau)}(N)$$

using dynamic programming (DP).

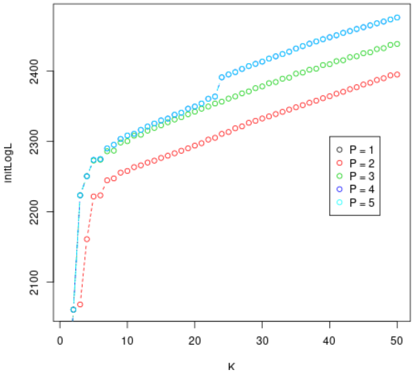
In practice

- Need to run EM-DP for

$$1 \leqslant K \leqslant K_{\max}, \quad 1 \leqslant P \leqslant P_{\max}$$

- With few guaranty to attain the global optimum (very few EM iterations)

First rounds of EM-DP



In practice

- Need to run EM-DP for

$$1 \leqslant K \leqslant K_{\max}, \quad 1 \leqslant P \leqslant P_{\max}$$

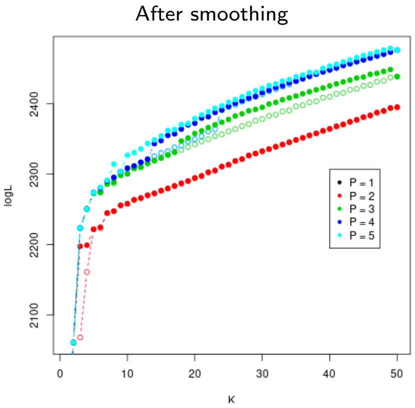
- With few guaranty to attain the global optimum (very few EM iterations)

- Need for (time-consuming) smoothing steps, restarting EM-DP
  $\rightarrow$ with neighbor mixture parms:

$$(\widehat{\pi}_{K-1,P}, \widehat{\lambda}_{K-1,P}, \widehat{\tau}_{K,P})$$
$$(\widehat{\pi}_{K+1,P}, \widehat{\lambda}_{K+1,P}, \widehat{\tau}_{K,P})$$

  $\rightarrow$ or neighbor segmentation parms:

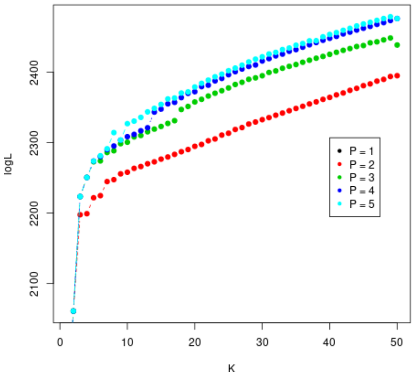
$$(\widehat{\pi}_{K,P}, \widehat{\lambda}_{K,P}, \widehat{\tau}_{K,P-1})$$
$$(\widehat{\pi}_{K,P}, \widehat{\lambda}_{K,P}, \widehat{\tau}_{K,P+1})$$



Model selection

- Need to select both K and P
- Cross-validation too demanding

$\log p_{K,P}(N)$



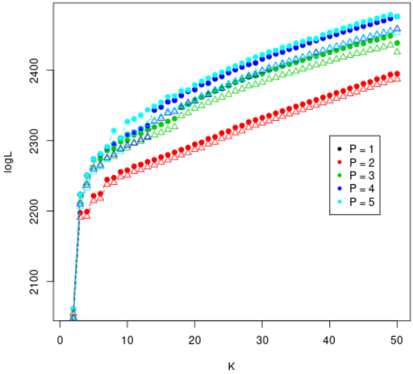
Model selection

- Need to select both K and P
- Cross-validation too demanding
- BIC penalty for mixture

$$pen(P) = (2P - 1) \log(K)/2$$

(wrong: see [LMH06])

$BIC_{\text{mixture}}(K, P)$



Model selection

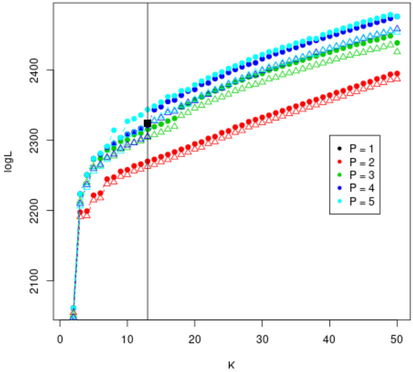
- Need to select both K and P
- Cross-validation too demanding
- BIC penalty for mixture

$$pen(P) = (2P - 1)\log(K)/2$$

(wrong: see [LMH06])

- Penalty for segmentation: slope heuristic (capushe)

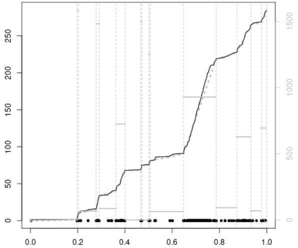
$\widehat{K} = 13, \widehat{P} = 5$



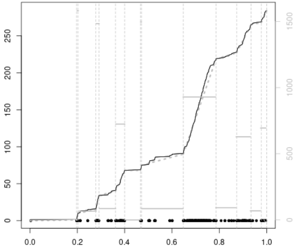
Some examples

Segmentation (DP)

$\widehat{K}_{DP} = 16$

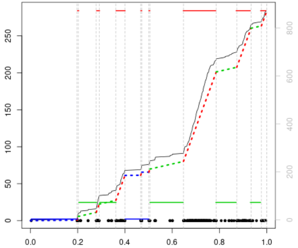


$\widehat{K}_{EMDP} = 14$

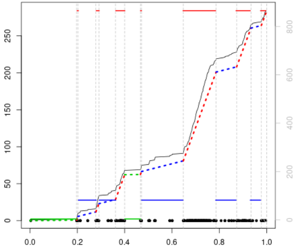


Segmentation clustering (EM-DP)

$\widehat{K}_{DP} = 16, \ \widehat{P}_{EMDP} = 3$



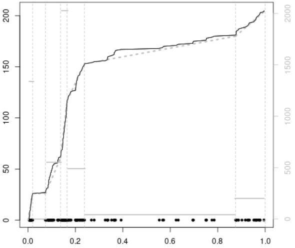
$\widehat{K}_{EMDP} = 14, \ \widehat{P}_{EMDP} = 3$



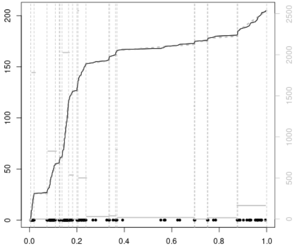
Some examples

Segmentation (DP)

$\widehat{K}_{DP} = 7$

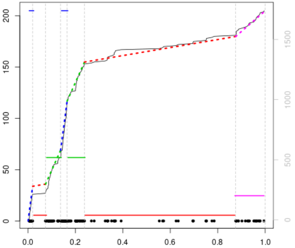


$\widehat{K}_{EMDP} = 23$

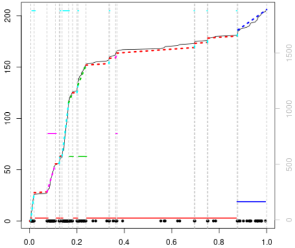


Segmentation clustering (EM-DP)

$\widehat{K}_{DP} = 7, \ \widehat{P}_{EMDP} = 5$



$\widehat{K}_{EMDP} = 23, \ \widehat{P}_{EMDP} = 5$



To summarize (1/2)

Segmentation-clustering allows to classify segment according to underlying behaviors

To summarize (1/2)

Segmentation-clustering allows to classify segment according to underlying behaviors

What does work.

- Segmentation step still efficient thank to the concavity property
- Fast (!) EM-DP algorithm for the whole inference

To summarize (1/2)

Segmentation-clustering allows to classify segment according to underlying behaviors

What does work.

- Segmentation step still efficient thank to the concavity property
- Fast (!) EM-DP algorithm for the whole inference

What does not work.

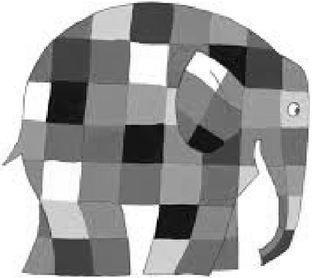
- Exploration of (K, P) computationaly demanding
- Need for a dedicated model selection procedure

To summarize (2/2)

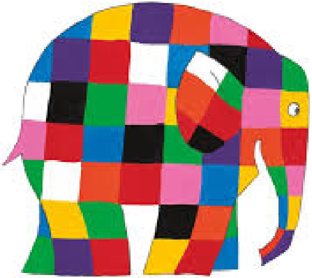
efficiEnt cLustering and segMEntation of Poisson pRocesses

To summarize (2/2)

efficiEnt cLustering and segMEntation of Poisson pRocesses



$\longrightarrow$



References

C Dion-Blanc, E Lebarbier, and S Robin. Multiple change-point detection for Poisson processes. Technical Report 2302.09103, arXiv, 2023.

E. Lebarbier and T. Mary-Huard. Une introduction au critère BIC : fondements théoriques et interprétation. *J. Soc. Française Statis.*, 147(1):39–57, 2006.

F. Picard, S. Robin, E Lebarbier, and J-J Daudin. A segmentation/clustering model for the analysis of array CGH data. *Biometrics*, 63(3):758–766, 2007.