Change-point detection in regression models for ordered data via the max-EM algorithm

Modibo Diabaté, Grégory Nuel, Olivier Bouaziz

Change-point detection in regression models for ordered data via the max-EM algorithm

Modibo Diabaté[1], Grégory Nuel[2] and Olivier Bouaziz[1]

[1]Université Paris Cité, CNRS, MAP5, F-75006 Paris, France
[2]LPSM (UMR CNRS 8001), Sorbonne Université, France

Abstract

We consider the problem of breakpoint detection in a regression modeling framework. To that end, we introduce a novel method, the max-EM algorithm which combines a constrained Hidden Markov Model with the Classification-EM (CEM) algorithm. This algorithm has linear complexity and provides accurate breakpoints detection and parameter estimations. We derive a theoretical result that shows that the likelihood of the data as a function of the regression parameters and the breakpoints location is increased at each step of the algorithm. We also present two initialization methods for the location of the breakpoints in order to deal with local maxima issues. Finally, a statistical test in the one breakpoint situation is developed. Simulation experiments based on linear, logistic, Poisson and Accelerated Failure Time regression models show that the final method that includes the initialization procedure and the max-EM algorithm has a strong performance both in terms of parameters estimation and breakpoints detection. The statistical test is also evaluated and exhibits a correct rejection rate under the null hypothesis and a strong power under various alternatives. Two real dataset are analyzed, the UCI bike sharing and the health disease data, where the interest of the method to detect heterogeneity in the distribution of the data is illustrated.

Keywords: breakpoint detection, CEM, constrained HMM, regression modeling, maximum likelihood inference, statistical breakpoint test.

1 Introduction

Breakpoint modeling is a major topic in many applications and taking them into account generally allows a better understanding of the studied problem. In finance, the detection of points of variation in time series of volatility of financial assets allows a better appreciation of the market risks and represents a subject of great interest [see 1, 2, 3]. Other examples include environmental changes over time [see 4, 5, 6] or speech perception in the analysis of sound signals [see 7, 8]. It is also an important and relevant topic in many medical applications, as the study of breakpoint detection allows to detect heterogeneity in patients data: this is particularly interesting in personalized medicine where the goal is to optimize treatment strategies. Applications of breakpoint models also include genomic data in cancer studies like in [9, 10, 11, 12] where the efficient detection of the change in the number of DNA copies in cancer data makes it possible to detect the presence of cancer cells (characterized by a faster division frequency), or even to study the progression and type of a cancerous tumor. Several approaches have been proposed to deal with such problems and breakpoint detection methods can be separated in two main classes: exact breakpoint calculation and statistical methods. In the first case, the aim is to develop an efficient algorithm that exhaustively explores all possible segmentations (corresponding to all possible breakpoints) while in the second case, the aim is to build a statistical model that aims at finding the most probable segmentation.

Exact calculation of breakpoints can be performed using dynamic programming with the Optimal Partitioning (OP) approach [see 13]. However, this method has a high computational complexity of order $O(n^2)$ which makes it intractable to use with large datasets. Optimized versions of this dynamic algorithm involving a pruning step have been proposed to reduce the algorithmic complexity. In particular, the Pruned Exact Linear Time (PELT) method introduced by [4] has a linear computational cost when the number of change-points increases as we observe more data. Many other algorithms have been introduced to attempt to reduce the time complexity of this algorithm. This is the case for instance of the Functional Pruning Optimal Partitioning (FPOP) algorithm [see 14, 15] and its extension, the Generalized Functional Pruning Optimal Partitioning (GFPOP) algorithm [see 16, 17]. These algorithms have the property that they can include constraints and they can consider a wide range of loss functions. See [17] for a more detailed review of the dynamic programming based algorithms that were developed for breakpoint detection. However, all these methods are not suited to deal with regression modeling. They are tailored to the detection of breakpoints over a series of values of a response vector but they cannot include information from a covariate matrix. Also, in the simple mean model, where the differences in terms of segments is characterized by the mean of the response vector, the gfpop algorithm can only work under homoscedasticity.

In this work, we present a general approach based on statistical models that extends the dynamic programming algorithms to regression modeling but is no longer based on exact breakpoints calculation. The main challenge is then to be able to extend the breakpoint detection to more general models while also keeping a good accuracy in breakpoint detection. In [18] and [19], the authors have proposed a methodology that combines Hidden Markov Model (HMM) methods and the Expectation maximization (EM) algorithm to achieve this goal, in a logistic and a Cox regression models, respectively. While the method has shown to be of interest to detect heterogeneity in binary or time to event data, it also suffers two major drawbacks. First, the algorithm is highly sensitive to the initialization value of the parameters, where several initialization choices may lead to different breakpoints and estimated parameters. Second, if the focus is mostly on breakpoint detection, the EM step is not adapted. This is because it makes a compromise by finding the most relevant regression parameters that maximize the averaged likelihood over all possible segmentations when the same value of the regression parameter is used in each segmentation. This lead us to the development of a new method, called the max-EM algorithm. In this method, the EM step is replaced by a Classification EM (CEM) step, inspired from the work of [20]. Moreover, the segments are modeled using HMM, as in [18], but we introduce a new forward-backward algorithm where the computation of the forward and backward quantities is performed by taking the maximum (instead of the sum) over a sequence of segments. We show that this new algorithm is well suited to the breakpoint detection problem where the aim is to find the best segmentation among a fix number of segments in a general regression framework. Then, we also present two strategies for the initialization of this iterative algorithm. The first one is based on the Fused Lasso (FL) method [see 21, 22] where we implement the overparameterized setting with a number of segments equal to the number of individuals and we penalize the values of regression parameters over two consecutive segments. The second one is based on Binary Segmentation (BS) where the idea is to recursively apply the simple one breakpoint model [see 23]. Both approaches allow to derive a sequence of breakpoint candidates. From these, we run the max-EM algorithm for all possible combinations and keep the result from the model with the highest likelihood value. Finally, we address the problem of heterogeneity detection from a statistical point of view. More precisely, we develop a new statistical test in the one breakpoint situation. From a theoretical point of view, the derivation of the distribution of the statistical test is extremely difficult due to the fact that it involves the maximum over all possible segmentations of the maximum over all parameter values. This is why we derive asymptotic approximations of the likelihood ratio test from which the maximum over all possible segmentations can be easily computed. This provides a very useful and

easily implementable statistical test. Our simulation results show that the max-EM algorithm works well in practice, both for the detection of breakpoints and the estimation of regression parameters. We observe that the initialization procedures find relevant breakpoints that allow to stabilize the results with an advantage over the BS initialization in terms of performance and computation time balance. Regarding the statistical test, we observed that it is well calibrated under the null hypothesis and has a strong power under various alternatives.

The paper is organized as follows. We first present the main goals of the paper in the next section. Then, the EM algorithm combined with HMM is recalled in Section 3. We show in particular that it does not address the problem of breakpoint detection. We further introduce our new max-EM algorithm, we derive its theoretical properties and the two initialization procedures are presented. We conclude the section by presenting a standard Bayesian Information Criterion (BIC) used to select the number of breakpoints. In Section 4, we present the approximation formulas for the statistical test based on likelihood ratio computation. In Section 5 extensive simulation experiments are conducted: the performance of our method for breakpoints detection and parameters estimation is studied through several regression modeling (linear, logistic, Poisson and AFT regressions) and different number of breakpoints (from 1 to 5). The statistical test is also studied under the same regression models. In Section 6 we study two real dataset using our new method: the UCI bike sharing dataset where the aim is to detect change of trends with respect to the date for the number of total daily rental bikes and the UCI heart disease dataset where the aim is to detect heterogeneity in the effect of fasting blood sugar on the risk of developing a heart disease.

2 Objectives

We consider a maximum likelihood based problem in the situation where the distribution of the data depends on K segments. More specifically, we assume there exists $K-1$ breakpoints $(n_1^*, \dots, n_{K-1}^*) \in \{1, \dots, n-1\}$ such that $n_0^* = 0 < n_1^* < \cdots < n_{K-1}^* < n_K^* = n$ and for $k = 1, \dots, K$, $X_{n_{k-1}^*+1}, \dots, X_{n_k^*}$ are independent and identically distributed (iid) following a distribution with continuous/discrete probability distribution function, denoted $e_i(k; \theta_k^*)$, that depends on an unknown d dimensional parameter $\theta_k^* \in \Theta \subset \mathbb{R}^d$. Importantly, the number and location of the segments are also assumed to be unknown. Let $R_i \in \{1, \dots, K\}$ be the latent variable representing the segment index associated to each individual: $R_i = k$ for $i \in \{n_{k-1}^*+1, \dots, n_k^*\}$. Using this notation, $e_i(k; \theta_k) = \mathbb{P}(X_i \mid R_i = k; \theta_k)$ represents the conditional distribution of X_i given $R_i = k$, evaluated at the parameter θ_k. For a given set of breakpoints and parameters, the log-likelihood of such a model can be written as:

$$\begin{aligned} \ell_n(\boldsymbol{\theta}; n_{1:(K-1)}) &= \log\big(\mathbb{P}(X_{1:n}, R_{1:n} \mid \boldsymbol{\theta})\big) \\ &= \sum_{k=1}^{K} \sum_{i \in \mathcal{C}_k} \log\big(e_i(k; \theta_k)\big) + \log\big(\mathbb{P}(R_{1:n})\big), \end{aligned} \tag{1}$$

where $\mathcal{C}_k = \{n_{k-1}+1, \dots, n_k\}$ and we use the compact notations $X_{1:n}$, $R_{1:n}$, $\boldsymbol{\theta}$, $n_{1:(K-1)}$ to represent the set of variables and parameters $X_1, \dots, X_n$, $R_1, \dots, R_n$, $\theta_1, \dots, \theta_K$, $n_1, \dots, n_{K-1}$ respectively. It should be noted that ℓ_n corresponds to the CML criterion (C_1) introduced in [20], since the $\mathbb{P}(R_{1:n})$ term can be omitted in the maximization. However, the major difference with their criterion comes from the structure of the $\mathcal{C}_k$ sets which can only contain ordered values of individuals in our case. In order to take into account this order of the individuals, we impose a Markov structure upon the R_i's: we assume that each R_i only depends on R_{i-1}, $i = 2, \dots, n$. We also impose that $\mathbb{P}(R_1 = 1) = 1$ and we restrict our analysis to the set of Markov chains verifying $R_n = K$.

In practice, the interest of the method lies in the regression modeling of joint distributions, such that $X_i = (Y_i, Z_i)$, where Y_i is an outcome variable and Z_i a covariate vector of dimension

d. Typically the conditional distribution of the Y_i's given the Z_i's will depend on $\theta_1^*, \ldots, \theta_K^*$ while the marginal distribution of the Z_i's will be parameter free. In this regression framework, the conditional density of X_i given $R_i = k$, $e_i(k; \theta_k)$, can be directly specified as following a regression model. In particular, in the simulation section, we consider the linear, the logistic, the Poisson and the Accelerated Failure Time (AFT) regression models.

2.1 First goal

The first goal of this paper is to develop a method for inferring the number and locations of the segments along with the estimation of the parameters θ_k. This is done, when the number of breakpoints is fixed, by maximizing Equation (1) with respect to both the n_k's and θ_k's

$$\max_{n_1,\ldots,n_{K-1}} \sup_{\theta_1,\ldots,\theta_K} \ell_n(\boldsymbol{\theta}; n_{1:(K-1)}) \tag{2}$$

This maximization problem can be directly solved sequentially by computing the maximum of $\ell_n(\theta_{1_K}; n_{1:(K-1)})$ with respect to θ_k for each $\mathcal{C}_k$, and then by taking the maximum of all these values. This naive approach will be called the "Brute force" algorithm in the following. It will accurately detect the breakpoints and the parameter values and is very simple to implement. However, the computation of our log-likelihood criterion for all possible segmentations is computationally very intensive ($O(n^{K+1})$ for the problem with K breakpoints) and it is therefore not a feasible approach for large datasets or for several number of segments.

As an alternative, one can use the EM algorithm to take into account the latent segment index. Models based on the EM algorithm and constrained Hidden Markov Model (HMM) were proposed in [18] and [19]. Those methods are fast to execute (linear complexity) and provide high accuracy when properly initialized. However, we show in Section 3.1 that the EM method does not solve the problem in Equation (2). Instead, it attempts to find the $\boldsymbol{\theta}$ parameter that makes the best comprise when we average all possible segmentations and the same value of $\boldsymbol{\theta}$ is used in each segmentation. This is why we introduce, in Section 3.2, a novel method, called the max-EM algorithm, and show in Section 3.3 that this max-EM algorithm is well adapted to the maximization problem of Equation (2) in the sense that each iteration of the algorithm is shown to increase the log-likelihood. As the algorithm is highly sensitive to parameters initialization, we also develop two different strategies for the initialization of the max-EM algorithm in Section 3.4. Since the max-EM algorithm only works for a fix value of K we also propose, in Section 3.5, an heuristic based on the Bayesian Information Criterion (BIC) to infer the number of breakpoints K. The final max-EM algorithm, integrating the proposed initialization strategy, is implemented and evaluated on simulated data in Section 5. Various regression models and number of breakpoints are considered. In the one breakpoint setting, our method is compared with the "Brute force" algorithm. In the absence of covariates, our approach is compared with the optimal dynamic programming algorithm GFPOP [see 17] when a simple mean model is considered. All our results show that our method works well in practice and can extend the GFPOP method to regression modeling.

2.2 Second goal

The second goal of this paper is to develop a new statistical test in the one breakpoint scenario. In other words we propose a statistical test to make a decision between the two hypothesis

$$\begin{aligned}
&(H_0) : X_1, \ldots, X_n \sim \mathcal{L}(\cdot, \theta^*) \text{ with } \theta^* \in \Theta \\
&(H_1) : \exists\, n_1^* \in \{2, \ldots, n-1\} : X_1, \ldots, X_{n_1^*} \sim \mathcal{L}(\cdot, \theta_1^*), X_{n_1^*+1}, \ldots, X_n \sim \mathcal{L}(\cdot, \theta_2^*), \\
&\qquad \text{with } \theta_1^* \neq \theta_2^*, (\theta_1^*, \theta_2^*) \in \Theta^2.
\end{aligned} \tag{3}$$

A likelihood based ratio test is presented in Section 4 for this purpose. The statistical test requires to take the maximum of the log-likelihood ratio over all possible values of the breakpoint

n_1^*, and for each value of n_1^*, over all possible values of the regression parameters. Deriving the exact or asymptotic distribution of this statistical test is extremely challenging. This is why we instead provide, in Section 4, an approximation formula of the log-likelihood ratio for any breakpoint value. The interest in this approximation formula lies in the fact that the $\boldsymbol{\theta}$ parameter and the Hessian matrix need only to be estimated under the null hypothesis. The score vector is also computed at the $\boldsymbol{\theta}$ parameter estimated under the null hypothesis but evaluated on the two segments. As a result, computing this approximated formula for all possible breakpoint values is extremely fast. In practice, this allows to easily test for breakpoint detections when using regression modeling. We also show in Section 5.3 that the approximation formula works well on simulated data: under various regression models and breakpoint situations, we observe that using our formula the statistical test has the correct rejection rate under the null hypothesis and a good power under various alternative hypothesis.

3 Breakpoint detection methodology

In this section, we present our approach based on the max-EM algorithm to perform breakpoint detection in ordered data. The max-EM approach is based on the use of a constrained HMM via a forward-backward type algorithm inspired from the EM algorithm. In Section 3.1, we first recall the EM method presented in [18] and explain why this method does not maximize the criterion defined in Equation (1). We then introduce the max-EM algorithm in Section 3.2 and show in Section 3.3 that each iteration of the algorithm increases the likelihood in Equation (1). In Section 3.4 we propose two different strategies for the initialization of the algorithm. In Section 3.5, we explain how the choice of the number of breakpoints K can be done based on the Bayesian Information Criterion (BIC).

3.1 Review on the EM algorithm for ordered data in a HMM

The EM algorithm is an iterative method designed to maximize the observed likelihood $\mathbb{P}(X_{1:n} \mid \boldsymbol{\theta})$. Given a current parameter $\boldsymbol{\theta}^{(m)}$, the E-step is based on the computation of the quantity

$$\mathbb{Q}(\boldsymbol{\theta} \mid \boldsymbol{\theta}^{(m)}) = \mathbb{E}\left[\log \mathbb{P}(X_{1:n}, R_{1:n} \mid \theta) \mid X_{1:n}; \boldsymbol{\theta}^{(m)}\right] = \sum_{R_{1:n}} \mathbb{P}(R_{1:n} \mid X_{1:n}; \boldsymbol{\theta}^{(m)}) \log \mathbb{P}(X_{1:n}, R_{1:n} \mid \boldsymbol{\theta}),$$

where the sum is taken over all possible segmentations such that $R_n = K$. Introduce the weights

$$\begin{aligned}\omega_i(k; \boldsymbol{\theta}^{(m)}) &= \mathbb{P}(R_i = k \mid X_{1:n}, R_n = K; \boldsymbol{\theta}^{(m)}) \\ &= \frac{\mathbb{P}(X_{1:i}, R_i = k \mid \boldsymbol{\theta}^{(m)}) \mathbb{P}(X_{(i+1):n}, R_n = K \mid R_i = k; \boldsymbol{\theta}^{(m)})}{\mathbb{P}(X_{1:n}, R_n = K)}.\end{aligned}$$

It has been proved in [18] (see Supporting material) that

$$\mathbb{Q}(\boldsymbol{\theta} \mid \boldsymbol{\theta}^{(m)}) = \sum_{i=1}^{n} \sum_{k=1}^{K} \omega_i(k; \boldsymbol{\theta}^{(m)}) \log\big(e_i(k; \theta_k)\big).$$

The E-step can therefore be implemented after computation of the ω_i's. This is achieved by means of a forward-backward algorithm.

By setting $F_i(k; \boldsymbol{\theta}) = \mathbb{P}(X_{1:i}, R_i = k \mid \boldsymbol{\theta})$, for $i = 1, \ldots, n$ (the so-called forward quantities) and $B_i(k; \boldsymbol{\theta}) = \mathbb{P}(X_{(i+1):n}, R_n = K \mid R_i = k; \boldsymbol{\theta})$, for $i = 1, \ldots, n-1$ (the so-called backward quantities), we then have

$$\omega_i(k; \boldsymbol{\theta}) = \frac{F_i(k; \boldsymbol{\theta}) B_i(k; \boldsymbol{\theta})}{\mathbb{P}(X_{1:n}, R_n = K)}.$$

The forward and backward quantities can be recursively computed as follows, for $i = 2, \dots, n, k = 1, \dots, K$:

$$\begin{aligned} F_1(k;\boldsymbol{\theta}) &= e_1(1;\theta_1)\mathbb{1}_{k=1}, \\ B_n(k) &= \mathbb{1}_{k=K}, \\ F_i(k;\boldsymbol{\theta}) &= \sum_{j=k-1}^{k} F_{i-1}(j;\boldsymbol{\theta})\phi_i(j,k;\boldsymbol{\theta})\mathbb{1}_{j\geq 1}, \\ B_{i-1}(k;\boldsymbol{\theta}) &= \sum_{j=k}^{k+1} \phi_i(k,j;\boldsymbol{\theta})B_i(j;\boldsymbol{\theta})\mathbb{1}_{j\leq K}, \end{aligned}$$

where

$$\phi_i(j,k;\boldsymbol{\theta}) = \mathbb{P}(R_i = k, X_i \mid R_{i-1} = j;\boldsymbol{\theta}) = e_i(k;\theta_k)\mathbb{P}(R_i = k \mid R_{i-1} = j).$$

In practice, these calculations are done in logarithmic scale in order to avoid underflow problems (see Appendix A.3.1 for more details).

To summarize, the EM algorithm follows the two steps:

- **E:** computation of the weights $\omega_i(k;\boldsymbol{\theta}^{(m)}) = \mathbb{P}(R_i = k \mid X_{1:n};\boldsymbol{\theta}^{(m)}) \propto F_i(k,\boldsymbol{\theta}^{(m)})B_i(k,\boldsymbol{\theta}^{(m)})$ (use of the forward-backward algorithm).

- **M:** update of the parameter value: $\boldsymbol{\theta}^{(m+1)} = \underset{\boldsymbol{\theta}}{\operatorname{argmax}}\ \mathbb{Q}(\boldsymbol{\theta} \mid \boldsymbol{\theta}^{(m)})$.

As mentioned earlier, it is important to stress that this algorithm maximizes with respect to $\boldsymbol{\theta}$ the observed likelihood

$$\mathbb{P}(X_{1:n} \mid \boldsymbol{\theta}) = \sum_{R_{1:n}} \mathbb{P}(X_{1:n} \mid R_{1:n};\boldsymbol{\theta})\mathbb{P}(R_{1:n}), \tag{4}$$

where the sum is taken over all possible segmentations such that $R_n = K$. Taking the logarithm of this quantity clearly gives a different expression than the objective quantity defined in Equation (1) and the EM algorithm will not provide a maximizer of $\ell_n(\boldsymbol{\theta}; n_{1:(K-1)})$. Looking at Equation (4), we see that the EM algorithm makes a compromise by finding the $\boldsymbol{\theta}$ parameter that maximizes the likelihood over all possible segmentations when $\boldsymbol{\theta}$ is shared in all segmentations.

3.2 The max-EM algorithm for ordered data in a HMM

Instead of averaging over all possible segmentations, the max-EM attempts at finding the best possible segmentation and at maximizing the θ_k parameter in each of these segments. For that purpose, we consider the max-forward and max-backward quantities that are given, for all $k \in \{1, \dots, K\}$, by

$$\begin{aligned} F_i^{\max}(k;\boldsymbol{\theta}) &= \max_{R_1,\dots,R_{(i-1)}} \mathbb{P}(R_{1:(i-1)}, R_i = k, X_{1:i} \mid \boldsymbol{\theta}), \text{ for } i = 1, \dots, n, \\ B_i^{\max}(k;\boldsymbol{\theta}) &= \max_{R_{i+1},\dots,R_{n-1}} \mathbb{P}(R_{(i+1):(n-1)}, R_n = K, X_{(i+1):n} \mid R_i = k;\boldsymbol{\theta}), \text{ for } i = 1, \dots, n-1, \end{aligned}$$

respectively. One should note the similarity with the forward and backward quantities introduced in the previous section where the sum symbol has been replaced by the maximum.

Furthermore, the max-forward and max-backward quantities can also be explicitly computed using the recurrence formulas:

$$F_i^{\max}(k;\boldsymbol{\theta}) = \max_{j\in\{k-1,k\}} F_{i-1}^{\max}(j;\boldsymbol{\theta})\phi_i(j,k;\boldsymbol{\theta}), \text{ for } i=2,\dots,n, k=2,\dots,K,$$
$$B_{i-1}^{\max}(k;\boldsymbol{\theta}) = \max_{j\in\{k,k+1\}} B_i^{\max}(j;\boldsymbol{\theta})\phi_i(k,j),, \text{ for } i=2,\dots,n-1, k=1,\dots,K-1,$$

with similar formulas for $F_1^{\max}(k;\boldsymbol{\theta})$, $F_i^{\max}(1;\boldsymbol{\theta})$, $B_{n-1}^{\max}(k;\boldsymbol{\theta})$, $B_{i-1}^{\max}(K;\boldsymbol{\theta})$ as in the previous section. Given a current parameter $\boldsymbol{\theta}^{(m)}$, the quantities $F_i^{\max}(k;\boldsymbol{\theta}^{(m)})$ and $B_i^{\max}(k;\boldsymbol{\theta}^{(m)})$ are then combined to compute the Maximum a Posteriori (MAP):

$$F_i^{\max}(k;\boldsymbol{\theta}^{(m)})B_i^{\max}(k;\boldsymbol{\theta}^{(m)}) = \max_{R_{1:(i-1)},R_{(i+1):(n-1)}} \mathbb{P}(R_{1:(i-1)}, R_i=k, R_{(i+1):(n-1)}, X_{1:n}, R_n=K \mid \boldsymbol{\theta}^{(m)}), \tag{5}$$

and from the MAP, we update the segmentation allocation as:

$$R_i^{\max(m+1)} = \operatorname*{argmax}_k F_i^{\max}(k;\boldsymbol{\theta}^{(m)})B_i^{\max}(k;\boldsymbol{\theta}^{(m)}).$$

Then, in order to update the value of the parameter $\boldsymbol{\theta}$, we maximize, with respect to $\boldsymbol{\theta}$, the quantity

$$\sum_{k=1}^{K}\sum_{i=1}^{n} \log e_i(k;\theta_k)\mathbb{1}_{R_i^{\max(m+1)}=k}. \tag{6}$$

Note that, in the above formula, the maximization can be performed for each θ_k separately by splitting the log likelihood over each segment. The max-forward and max-backward quantities thus lead to the so-called max-EM algorithm. To summarize, its E- and M-steps proceed as follows:

- **E-step:**
  - Computation of $F_i^{\max}(k;\boldsymbol{\theta}^{(m)})$ and $B_i^{\max}(k;\boldsymbol{\theta}^{(m)})$, for $i=1,\dots,n$.
  - Update of the segmentation allocation
$$R_i^{\max(m+1)} = \operatorname*{argmax}_k F_i^{\max}(k;\boldsymbol{\theta}^{(m)})B_i^{\max}(k;\boldsymbol{\theta}^{(m)}),\ i=1,\dots,n.$$
- **M-step:** update of the parameter value
$$\boldsymbol{\theta}^{(m+1)} = \operatorname*{argmax}_{\boldsymbol{\theta}} \sum_{k=1}^{K}\sum_{i=1}^{n} \log e_i(k;\theta_k)\mathbb{1}_{R_i^{\max(m+1)}=k}.$$

Even though underflow issues are less problematic with the max-forward max-backward algorithm, those situations can still arise in practice. The logarithmic scaling is done in a very similar way as in the previous forward-backward algorithm (see Appendix A.3.2 for more details).

3.3 Convergence properties of the max-EM algorithm

In the next proposition we show that each iteration of the max-EM algorithm increases the log-likelihood ℓ_n defined in Equation (1). The proof is deferred to the Appendix section and is based on the proof from [20]. The main difference in our proof comes from the structure of the data where the individuals are ordered and the segment indexes that are assumed to follow a HMM. We equivalently denote $(\boldsymbol{\theta}^{(m)}, \boldsymbol{\mathcal{C}}^{(m)})$ or $(\boldsymbol{\theta}^{(m)}, n_{1:(K-1)}^{(m)})$ the parameters values obtained after the m^{th} step of the max-EM algorithm, where $\boldsymbol{\mathcal{C}}^{(m)} = (\mathcal{C}_1^{(m)},\dots,\mathcal{C}_K^{(m)})$ and we recall that $\mathcal{C}_k^{(m)} = \{n_{k-1}^{(m)}+1,\dots,n_k^{(m)}\}$ represents the set of individuals such that $R_i^{\max(m)} = k$.

Proposition 1. *The sequence of iterates* $(\boldsymbol{\theta}^{(m)}, \boldsymbol{\mathcal{C}}^{(m)})_{m\geq 1}$ *generated using the max-EM algorithm satisfies* $\ell_n(\boldsymbol{\theta}^{(m+1)}; n^{(m+1)}_{1:(K-1)}) \geq \ell_n(\boldsymbol{\theta}^{(m)}; n^{(m)}_{1:(K-1)})$. *Moreover, if for each set* $\{n^*_{k-1} + 1, \ldots, n^*_k\}$, $k = 1, \ldots, K$, *the associated log-likelihood* $\sum_{i\in\mathcal{C}^*_k} \log\big(e_i(k;\theta_k)\big)$ *has a unique maximum, then the sequence* $(\boldsymbol{\theta}^{(m)}, \boldsymbol{\mathcal{C}}^{(m)})_{m\geq 1}$ *converges towards a stationary parameter.*

3.4 Discussion on the algorithm initialization

The max-EM algorithm, like the standard EM algorithm and its variants, is sensitive to parameter initialization due to problems of convergence towards local maxima. When we have no information on the parameters value, it is advised to initialize these algorithms with several different initial values and analyze which initialization best maximizes the likelihood. In our setting, the aim is to define a set of $K'-1$ initialization values for the breakpoints, with $K' \geq K$. Once those values are found, we run our max-EM algorithm for all possible combinations of $K-1$ breakpoints among $K'-1$. For each of these combinations, we can start the max-EM algorithm by maximizing Equation (6) and then iterate the max-EM algorithm. Among all initializations, the final result is the one with maximum likelihood value.

One way to determine the set of breakpoints initializations is to randomly select them. In our experience, this strategy leads to inaccurate results even for simple problems with one or two breakpoints unless the number of breakpoints is very large, which, in turn, is problematic as the computation time drastically increases with the value of K'. The challenge is therefore to define efficient methods that provide good results with a small set of breakpoint initialization values. In the following, we propose two methods, the first one is based on the Fused-Lasso (FL) algorithm and the other one is based on Binary Segmentation (BS).

3.4.1 Fused-Lasso initialization

Our first approach uses the Fused-Lasso (FL) algorithm [see 21, 22] in the overparameterized model where the number of segments is equal to the number of individuals. The selection of candidate breakpoints goes through the following steps.

1. First, solve the problem

$$\boldsymbol{\theta} \in \underset{\theta_1,\ldots,\theta_n}{\arg\max} \left\{ \sum_{i=1}^{n} \log(e_i(i;\theta_i)) - \lambda \sum_{j=1}^{d} \sum_{i=1}^{n-1} |\theta^j_{i+1} - \theta^j_i| \right\},$$

   where $\lambda > 0$ is a penalty term and θ^j_i represents the jth component of the d-dimensional θ_i parameter. This is simply a penalized version of Equation (1) where $K = n$ and $\mathcal{C}_k = k$ for $k = 1, \ldots, n$. We implement this FL problem using the `glmnet` R package by rewriting it in terms of a standard Lasso problem through the parametrization $(\theta^1_1, \ldots, \theta^1_n, \ldots, \theta^d_1 \ldots, \theta^d_n)^\top = D\gamma$ where D is a block matrix of size $dn \times dn$ whose d diagonal blocks are equal to a lower triangular matrix with nonzero elements equal to 1 and whose $d^2 - d$ off-diagonal blocks are equal to matrices of zeros. See [24] for an example of such implementation of the FL algorithm.

2. With the `glmnet` R package, the problem is solved for a grid of λ values. Each of these values corresponds to a number of different θ parameters: when all θ^j_i are different from all θ^j_{i+1} parameters, we consider that the distribution of the data is different between the two segments. For a high penalty value, all θ^j_i are equal to all θ^j_{i+1}, for $i = 1, \ldots, n-1$ and there is only one segment. As the penalty value decreases, the number of segments increases. Based on this regularization path we choose the maximum value of λ that corresponds to a number of segments equal to at least $5(K-1)$ breakpoints (that is at least $5(K-1)+1$ segments).

3. We conclude by removing the breakpoints that are too close to each other. We set a minimum number of individuals per segment equal to 50 and as long as this criterion is not met, we sequentially remove breakpoints starting from the breakpoints that are the closest to each other. We also impose to keep at least $\lfloor \frac{3}{2}(K-1) \rfloor$ breakpoints in this final selection.

Once this step is finished, we end up with a set of $K'-1$ potential breakpoints for the initialization of the max-EM algorithm. We will then run our max-EM algorithm for all possible combinations of $K-1$ breakpoints among $K'-1$. This means our algorithm will be run $\binom{K'-1}{K-1}$ times. The threshold values $5(K-1)$ and $\lfloor \frac{3}{2}(K-1) \rfloor$ used in steps 2. and 3., respectively, are arbitrary and were chosen based on simulation experiments. They seem to provide a good compromise between the need to explore a large number of initializations and computer complexity. In our simulation experiments, this strategy was working with scenarios up to 7 breakpoints. Of note, the algorithmic complexity for FL is of order $\mathcal{O}(n^2)$ in our case, since the penalization is applied to $n-1$ consecutive differences. Also, the total computation time is sensitive to the type of regression modeling that is implemented (typically, a Poisson regression model is more computer intensive than a linear model).

3.4.2 Binary Segmentation initialization

Our second approach is based on the Binary Segmentation (BS) strategy [see 23]. The idea is based on a recursive splitting of the data and application of the max-EM algorithm in the one breakpoint situation. We start by running the one breakpoint max-EM algorithm, where the breakpoint is initialized at the middle of the sample. Once this is done we separately consider the two sub-samples made by the two segments and we apply twice the one breakpoint max-EM algorithm in each of those sub-samples. Again, the max-EM is initialized by setting the initial breakpoint as the middle value of the sub-sample. This recursion is applied four times which provides us with a total of $1+2+4+8=15$ breakpoints. As before, we run our max-EM algorithm for all possible combinations of $K-1$ breakpoints among 15. The number of recursions is arbitrary and is based on simulation experiments. It is important to stress that the one breakpoint max-EM algorithm is extremely fast to run, of order $\mathcal{O}(n)$, and the whole procedure needed to define our set of breakpoint initializations requires 15 calls of our one breakpoint model. While it might be possible to reduce this number when K is small, it is rather convenient in practice to simply fix this value. This gives a computational advantage of the BS initialization over FL.

On the other hand, the set of initial breakpoints will tend to be larger with the BS method than with the FL method, which will also impact the computation time of the two strategies.

3.5 Inferring the number of breakpoints K with BIC

The methodology developed so far works only for a fixed number of K. In this section, we propose to use the Bayesian Information Criterion (BIC) to infer this value, as in [18]. The criterion has the following form:

$$-2\ell_n(\hat{\boldsymbol{\theta}}; n_{1:(K-1)}) + d \times K \times \log(n),$$

where $\hat{\boldsymbol{\theta}}$ is the estimated parameter using our max-EM algorithm and $d \times K$ is the number of estimated parameters. We will choose the value of K that minimizes this criterion. In practice, this means that we will need to run our max-EM algorithm (including the initialization strategy) for a sequence of values for K in order to find the final model and estimated parameters.

4 Statistical test for the one breakpoint situation

In this section we provide a statistical test for the two hypothesis (3) in the one breakpoint scenario. For likelihood based methods, a simple statistical test is the likelihood ratio which is defined in the following way. Let $\ell_n^{H_0}$ be the log-likelihood under (H_0) and $\ell_n^{H_1}$ be the log-likelihood under (H_1), that is

$$\ell_n^{H_0} = \sup_{\theta} \tilde{\ell}_n(\theta) = \sup_{\theta} \left\{ \sum_{i=1}^{n} \log\left(\mathbb{P}(X_i;\theta)\right) \right\},$$
$$\ell_n^{H_1} = \max_{n_1} \sup_{\theta_1,\theta_2} \ell_n(\theta_1,\theta_2;n_1) = \max_{n_1} \sup_{\theta_1,\theta_2} \left\{ \sum_{i=1}^{n_1} \log\left(\mathbb{P}(X_i;\theta_1)\right) + \sum_{i=n_1+1}^{n} \log\left(\mathbb{P}(X_i;\theta_2)\right) \right\},$$

where we have introduced the notation $\tilde{\ell}_n$ to represent the likelihood in the no-breakpoint model. Note also that, for the sake of simplicity, the R_i term was dropped in the notation $\mathbb{P}(X_i;\theta_k)$ to denote the probability distribution function $\mathbb{P}(X_i \mid R_i = k;\theta_k)$, $k = 1, 2$. We also recall that our methodology works for discrete or continuous random variables. In the above equations, the supremum is taken over $\theta, \theta_1, \theta_2 \in \Theta$ and the maximum is taken over $n_1 \in \{1, \ldots, n-1\}$. The test statistic is then defined as $T_n = 2(\ell_n^{H_1} - \ell_n^{H_0})$.

For a fixed value of n_1, we define $(\hat{\theta}_1, \hat{\theta}_2) = \arg\max_{\theta_1,\theta_2} \ell_n(\theta_1,\theta_2;n_1)$ and $\hat{\theta}_0 = \arg\max_{\theta} \tilde{\ell}_n(\theta)$. It should be noted that $\hat{\theta}_1$ and $\hat{\theta}_2$ depend on the value of n_1, even though this does not appear in the notation for the sake of simplicity. The test statistic can then be rewritten as $T_n = \max_{n_1}\{2(\ell_n(\hat{\theta}_1,\hat{\theta}_2;n_1) - \tilde{\ell}_n(\hat{\theta}_0))\}$. In Theorem 1, the asymptotic distribution of $2(\ell_n(\hat{\theta}_1,\hat{\theta}_2;n_1) - \tilde{\ell}_n(\hat{\theta}_0))$ is provided under (H_0), when assuming that n_1 and $n - n_1$ converge towards infinity. In the following, we define the estimator of the Fisher information under (H_0):

$$\hat{I}(\hat{\theta}_0) = -\frac{1}{n} \sum_{i=1}^{n} \nabla^2 \log(\mathbb{P}(X_i;\hat{\theta}_0)),$$

and we use the notation $u^{\otimes 2} = u^\top u$.

Theorem 1. *Let $n, n_1 \in \mathbb{N}^*$, such that $n > n_1$ and $n_1 \to \infty$, $n - n_1 \to \infty$. Then, under standard assumptions for maximum likelihood theory,*

$$\begin{aligned}
&2(\ell_n(\hat{\theta}_1,\hat{\theta}_2;n_1) - \tilde{\ell}_n(\hat{\theta}_0)) \\
&\quad = \frac{n-n_1}{nn_1} \left[\left(\hat{I}(\hat{\theta}_0)\right)^{-1/2} \sum_{i=1}^{n_1} \nabla \log\left(\mathbb{P}(X_i;\hat{\theta}_0)\right) \right]^{\otimes 2} \\
&\qquad + \frac{n_1}{n(n-n_1)} \left[\left(\hat{I}(\hat{\theta}_0)\right)^{-1/2} \sum_{i=n_1+1}^{n} \nabla \log\left(\mathbb{P}(X_i;\hat{\theta}_0)\right) \right]^{\otimes 2} \\
&\qquad - \frac{2}{n} \left(\sum_{i=1}^{n_1} \nabla \log\left(\mathbb{P}(X_i;\hat{\theta}_0)\right) \right)^\top \left(\hat{I}(\hat{\theta}_0)\right)^{-1} \left(\sum_{i=n_1+1}^{n} \nabla \log\left(\mathbb{P}(X_i;\hat{\theta}_0)\right) \right) + o_{\mathbb{P}}(1).
\end{aligned}$$

This theorem can be used to compute the distribution of T_n under (H_0) when n_1 and $n - n_1$ are large. The approximation of $2(\ell_n(\hat{\theta}_1,\hat{\theta}_2;n_1) - \tilde{\ell}_n(\hat{\theta}_0))$ provided by the theorem can be computed for a sequence of n_1 values in an efficient way, then taking the maximum over this sequence will provide an approximation of T_n. It should be noted that the approximation does not depend on the estimators $\hat{\theta}_1$ and $\hat{\theta}_2$; only estimators in the no-breakpoint model must be computed. In practice, the estimator $\hat{\theta}_0$, the estimator of the Hessian matrix based on the whole sample and evaluated at $\hat{\theta}_0$, the estimator of the score vector $\nabla \log(\mathbb{P}(X_i;\hat{\theta}_0))$ for $i = 1, \ldots, n$ are fast to compute.

When n_1 or $n-n_1$ are small, the remainder term in the approximation will no longer be small and this approximation should not be used. The next theorem provides two new approximations for $2(\ell_n(\hat{\theta}_1, \hat{\theta}_2; n_1) - \tilde{\ell}_n(\hat{\theta}_0))$ corresponding to these two settings.

Theorem 2. *Let $n, n_1 \in \mathbb{N}^*$, such that $n > n_1$.*

1. *Under standard assumptions for maximum likelihood theory, if n_1 is fixed and $n \to \infty$ then*

$$2(\ell_n(\hat{\theta}_1, \hat{\theta}_2; n_1) - \tilde{\ell}_n(\hat{\theta}_0)) = 2\sum_{i=1}^{n_1} \left\{ \log\left(\mathbb{P}(X_i; \hat{\theta}_1)\right) - \log\left(\mathbb{P}(X_i; \hat{\theta}_0)\right) \right\} + o_{\mathbb{P}}(1).$$

2. *Under standard assumptions for maximum likelihood theory, if $n_1 \to \infty$ and $n - n_1$ converges towards a positive constant, then*

$$2(\ell_n(\hat{\theta}_1, \hat{\theta}_2; n_1) - \tilde{\ell}_n(\hat{\theta}_0)) = 2\sum_{i=n_1+1}^{n} \left\{ \log\left(\mathbb{P}(X_i; \hat{\theta}_2)\right) - \log\left(\mathbb{P}(X_i; \hat{\theta}_0)\right) \right\} + o_{\mathbb{P}}(1).$$

As opposed to Theorem 1, those results require the computation of $\hat{\theta}_1$ and $\hat{\theta}_2$. However, the idea is to use 1. of Theorem 2 for small values of n_1 (typically less than 100) and to use 2. of Theorem 2 for small values of $n - n_1$ (typically less than 100). We will therefore combine Theorems 1 and 2 to compute $\{2(\ell_n(\hat{\theta}_1, \hat{\theta}_2; n_1) - \tilde{\ell}_n(\hat{\theta}_0))\}$ for all values of n_1 and take the maximum to derive T_n. The proofs of those two theorems are provided in the Appendix section.

5 Simulations

In the following, we will evaluate the performance of our method in various simulation settings. In Section 5.1 we consider a simple mean model which allows comparisons of our method with the Brute-Force method (in the one breakpoint situation) and the GFPOP algorithm. In Section 5.2, three regression models are considered: a linear, a logistic and a survival models. In Section 5.3, the power of the statistical test developed in Section 4 is investigated in the three previous regression models with one breakpoint.

All the simulations are replicated on $J = 500$ samples. For $j = 1, \ldots, J$, $k = 1, \ldots, K$, let $\hat{\theta}_k^{(j)} = (\hat{\theta}_{k,1}^{(j)}, \ldots, \hat{\theta}_{k,d}^{(j)})^\top \in \mathbb{R}^d$ denote the estimate of the true parameter $\theta_k^* = (\theta_{k,1}^*, \ldots, \theta_{k,d}^*)^\top$ in segment k, obtained from the jth Monte Carlo sample. In order to assess the performance of this estimator, the Mean Squared Error (MSE) decomposed as the sum of the variance (VAR) and the squared bias BIAS^2, and the Mean Absolute Percentage Error (MAPE) are used as metrics. They are defined in the following way:

$$\begin{aligned}
\text{MSE}(\hat{\boldsymbol{\theta}}) &= \frac{1}{KJ}\sum_{j=1}^{J}\sum_{k=1}^{K}(\hat{\theta}_k^{(j)} - \theta_k^*)^\top(\hat{\theta}_k^{(j)} - \theta_k^*) \\
\text{BIAS}^2(\hat{\boldsymbol{\theta}}) &= \frac{1}{K}\sum_{k=1}^{K}(\bar{\hat{\theta}}_k - \theta_k^*)^\top(\bar{\hat{\theta}}_k - \theta_k^*) \\
\text{VAR}(\hat{\boldsymbol{\theta}}) &= \frac{1}{KJ}\sum_{j=1}^{J}\sum_{k=1}^{K}(\hat{\theta}_k^{(j)} - \bar{\hat{\theta}}_k)^\top(\hat{\theta}_k^{(j)} - \bar{\hat{\theta}}_k) \\
\text{MAPE}(\hat{\boldsymbol{\theta}}) &= \frac{1}{KJ}\sum_{j=1}^{J}\sum_{k=1}^{K}\sum_{l=1}^{d}\left|\frac{\hat{\theta}_{k,l}^{(j)} - \theta_{k,l}^*}{\theta_{k,l}^*}\right|,
\end{aligned}$$

where $\bar{\hat{\theta}}_k = \sum_j \hat{\theta}_k^{(j)}/J$. Contrary to the MSE, bias and variance, the MAPE metric takes into account the amplitude of the parameter values. On the other hand, the accuracy error of breakpoints detection is evaluated through the criterion:

$$\text{ACCE(bp)} = \frac{1}{n}\sum_{j=1}^{J}\sum_{i=1}^{n} \mathbb{1}_{\hat{R}_i^{(j)} \neq R_i},$$

where $\hat{R}_i^{(j)}$ is the estimated segment index for individual i in sample j and we recall that R_i is the true segment index for individual i. Therefore, this metric evaluates the proportion of individuals that are allocated the incorrect segment index.

5.1 Implementation of the max-EM algorithm in the mean model

In this section we consider the simple following model:

$$\text{for } k = 1, \ldots, K,\ i = n_{k-1}^* + 1, \ldots, n_k^*,\ Y_i = \theta_k^* + \varepsilon_i,$$

where $\varepsilon_i \sim \mathcal{N}(0, \sigma^{*2})$ and $\theta_k^* \in \mathbb{R}$. This is an homoscedastic model since the variance σ^{*2} is assumed to be equal for all K segments. The aim of this simulation setting is first, to compare the two proposed initialisations, the one based on the Fused Lasso (FS) and the other based on Binary Segmentation (BS) and second, to compare our implementations with the Brute Force method and with the GFPOP algorithm. For this second goal, the comparison with brute force can only be made in a one breakpoint situation (that is when $K = 2$) due to computational issues arising for $K \geq 3$. We consider two settings, one with one breakpoint ($K = 2$) and another setting with 5 breakpoints ($K = 6$).

- One breakpoint: $\theta_1^* = 10$, $\theta_2^* = 12$, $\sigma^* = 3$, $n = 500$ and $n_1^* = 345$.
- Five breakpoints: $\theta_1^* = 19$, $\theta_2^* = 23$, $\theta_3^* = 30$, $\theta_4^* = 35$, $\theta_5^* = 42$, $\theta_6^* = 37$, $\sigma^* = 5$, $n = 1,000$ and $n_1^* = 82$, $n_2^* = 333$, $n_3^* = 508$, $n_4^* = 701$, $n_5^* = 945$.

The results are presented in Table 1 where the MSE of the algorithms are provided along with its decomposition as the sum of the variance and the squared bias. The MAPE of the parameters and of the breakpoints values is also computed.

In the one breakpoint setting we first observe that all three methods (max-EM with BS initialization, GFPOP and Brute Force) have the same performance for the proposed metrics. In fact, the estimates for all $J = 500$ samples are identical. On the other hand, the max-EM with FL initialization provides very similar results: indeed, by looking more closely at the results, it turns out that, out of the 500 replications, there is only one sample where max-EM with FL initialization provides a different breakpoint than the other methods. For this breakpoint, it finds the breakpoint $\hat{n}_1 = 319$ with corresponding likelihood-value equal to $-2,498.02$, when all the other methods find the breakpoint $\hat{n}_1 = 343$ with corresponding likelihood-value equal to $-2,497.98$ (we recall that the true breakpoint is $n_1^* = 345$). The distribution of the estimated breakpoint based on all three methods is also provided in Figure 1. It shows that the algorithms are extremely accurate in terms of breakpoint detection in this setting. Finally, the MSE for the standard deviation of the residuals is equal to 0.00442 for both max-EM algorithms and for the Brute Force method. We have also compared the computation time of the whole method based on the two initializations, with a clear advantage of the max-EM with BS initialization which runs on average in 2.5 seconds over max-EM with FL initialization which runs on average in 4.5 seconds.

In the five breakpoint setting, all methods provide a very accurate estimation of the parameters based on all metrics. However, the max-EM algorithm with FL initialization tends to be less performant: its variance is twice as big as the variance of the other methods. This

highlights the fact that this method sometimes find a sequence of breakpoints that are far from the truth, a phenomenon that does not occur with max-EM with BS initialization and GFPOP whose performances are very similar according to all metrics. Finally, the MSE for the standard deviation of the residuals is equal to 0.00139 for the max-EM algorithm with FL initialization and to 0.00094 for the max-EM algorithm with BS initialization.

In light of these results, our algorithm max-EM with BS initialization seems to provide the best tradeoff between accuracy and speed, since its computational cost is linear. In the next simulations, we will only present the results for the BS initialization in the main text, the results for the FL initialization can be found in Supplementary Material.

	One bp		Five bp		
	max-EM(FL)	max-EM(BS)/GFPOP/BF	max-EM(FL)	max-EM(BS)	GFPOP
MSE($\hat{\boldsymbol{\theta}}$)	0.03668	0.03674	3.66324	1.65040	1.39877
BIAS^2($\hat{\boldsymbol{\theta}}$)	0.00012	0.00012	0.02502	0.01592	0.02778
VAR($\hat{\boldsymbol{\theta}}$)	0.03656	0.03662	3.63822	1.63449	1.37099
MAPE($\hat{\boldsymbol{\theta}}$)	0.01997	0.01998	0.08801	0.07906	0.07787
ACCE(bp)	0.00680	0.00675	0.02756	0.01567	0.01449

Table 1: Results in the simple homoscedastic mean model with two scenarios: the one and five breakpoint models. The Mean Squared Error (MSE) of the estimated mean parameters, decomposed as the variance (VAR) plus squared bias (BIAS^2) along with the MAPE of the estimated parameters and the ACCE of the estimated breakpoints are provided. The max-EM algorithm is compared with the GFPOP and brute force algorithms.

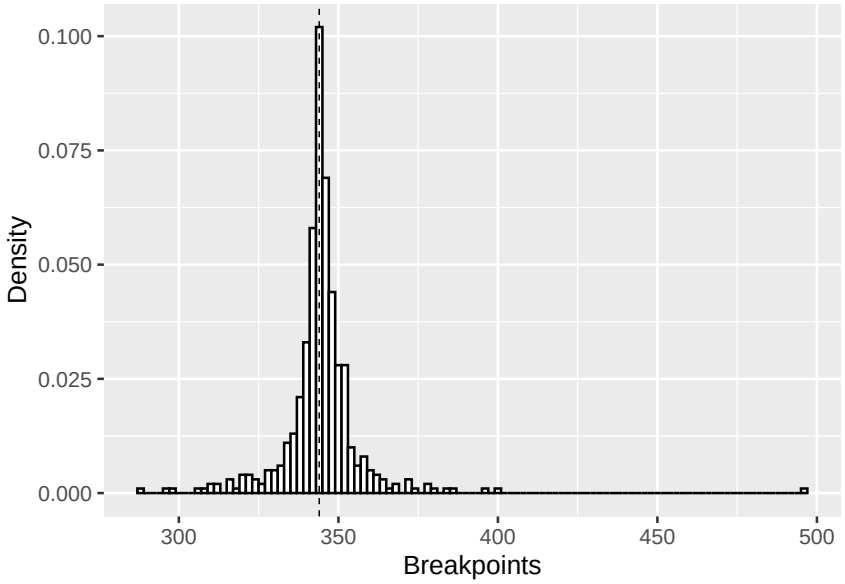

Figure 1: Distribution of breakpoints computed in the one breakpoint homoscedastic mean model. The distribution was obtained based on 500 replications and is identical for the max-EM with BS initialization, GFPOP and Brute Force algorithms. The vertical dotted line indicates the true breakpoint, equal to 345 in this simulation setting.

5.2 Implementation of the max-EM algorithm in regression models

In this section we consider three different regression models in different settings. A linear, a logistic and a survival regression models are studied based on scenarios with one and four breakpoints and several covariates. The models are described in details below.

- Model 1. Linear regression.

$$\text{For } k = 1, \dots, K, i = n^*_{k-1} + 1, \dots, n^*_k,\ Y_i = X_i^\top \theta^*_k + \varepsilon_i,$$

where $X_i = (1, X_{i,1}, X_{i,2})^\top$, $X_{i,1}$, $X_{i,2}$ are independent and follow a uniform distribution on $[0, 1]$ and ε_i follows a centered normal distribution with variance σ^2 equal to 6.25.

- Model 2. Logistic regression.

$$\text{For } k = 1, \dots, K, i = n^*_{k-1} + 1, \dots, n^*_k,\ \mathbb{P}[Y_i = 1 \mid X_i] = \frac{\exp(X_i^\top \theta^*_k)}{1 + \exp(X_i^\top \theta^*_k)},$$

where $X_i = (1, X_{i,1})^\top$ and $X_{i,1}$ follows a Bernoulli distribution with parameter $p = 0.5$.

- Model 3. Accelerated Failure Time/Cox regression.

$$\text{For } k = 1, \dots, K, i = n^*_{k-1} + 1, \dots, n^*_k,\ \log(Y_i) = X_i^\top \theta^*_k + \sigma\varepsilon_i,$$

where $X_i = (1, X_{i,1}, X_{i,2})^\top$, $X_{i,1}$, $X_{i,2}$ are independent and follow a uniform distribution on $[0, 1]$, ε_i has a probability density function equal to $f_\varepsilon(w) = \exp(w - \exp(w))$ and $\sigma \in \mathbb{R}$ is an extra scale parameter. In this model, the outcome Y_i is not directly observed but instead we observe the variable $T_i = Y_i \wedge C_i$, with C_i a censoring variable following an exponential distribution with parameter equal to 0.1 (that is with expectation equal to 10). With this censoring distribution, 35% of observations are censored on average. It is important to stress that even though this model is presented as an accelerated failure time model, it can also be recast into a Cox proportional hazard model [see 25]. Let $\lambda(\cdot \mid X_i)$ be the conditional hazard rate for the variable Y_i, then Model 3 is equivalent to assuming:

$$\text{for } k = 1, \dots, K,\ i = n^*_{k-1} + 1, \dots, n^*_k,\ \lambda(t \mid X_i) = \lambda_0(t) \exp(\tilde{X}_i^\top \beta^*_k),$$

where

$$\lambda_0(t) = \frac{1}{\sigma} \exp\left(-\frac{\theta^*_{k,1}}{\sigma}\right) t^{1/\sigma - 1},$$

$\tilde{X}_i = (X_{i,1}, X_{i,2})^\top$ and $\beta^*_k = -(1/\sigma)(\theta^*_{k,2}, \theta^*_{k,3})^\top$.

For each model, a one breakpoint ($K = 2$) and two breakpoint ($K = 3$) settings are considered. In the one breakpoint setting, all samples are of size $1,000$ and the breakpoints are equal to 553, 112 and 666 in the linear, logistic and survival models, respectively. In the two breakpoint setting, all samples are of size $1,000$ and the breakpoints are equal to 333 and 666 in the linear and logistic models, and to 375 and 689 in the survival model. The exact values of the parameters in each model and each breakpoint setting are provided in Table 2. The results from the max-EM algorithm with BS initialization are presented in Table 3. Some of the results with FL initialization can also be found in Supplementary Material. No competitors were computed in those simulation settings: the GFPOP algorithm cannot work with regression models and we were not able to implement the Brute Force algorithm due to computational issues. We observe a good performance of our method in all settings. In particular, the accuracy error of breakpoints detection, ACCE(bp), is extremely low in all settings, which implies that almost all individuals are assigned to the correct segment (the worst situation occurs for the logistic model with two breakpoints in which case ACCE(bp) equals 1.8%). Since the max-EM algorithm operates in two steps, with the segment allocation as the first step and separate parameters estimation in each segment as the second step, the parameters estimation error is mainly due to the performance of the maximum likelihood estimators inherent to each model and to the sample size in each segment. In the two breakpoint case, our estimator slightly

deteriorates in terms of MSE except for the logistic model. This is due to the balanced setting in terms of number of observations in each segment for the linear and survival models, while for the logistic model, the one breakpoint case is particularly unbalanced with few observations in the first segment (112 observations in the first segment and 888 observations in the second segment). By comparison, in the two breakpoint scenario, there are more observations in all three segments (333 in the first two segments and 334 in the third). Surprisingly, the survival model, that suffers from censoring and has the largest number of parameters, displays the best performance in terms of MSE and breakpoint detection, both in the one breakpoint and two breakpoint settings. In Table 1 of Supplementary Information, we observe that the FL initialization provides slightly better results than BS initialization for the linear and survival models, while for the logistic regression, BS initialization outperforms FL initialization except in terms of bias. In the two breakpoint situation, with the survival model, BS initialization has a slight advantage with all metrics except in terms of bias which is similar for the two initialization methods. Considering the computational advantage of BS initialization, those results are in favour of the BS initialization especially when the number of breakpoints is greater than one.

		One bp		Two bp		
		θ_1^*	θ_2^*	θ_1^*	θ_2^*	θ_3^*
Linear	Intercept	1.00	2.00	1.00	1.50	2.00
($\sigma = 2.5$)	cov. effect 1	11.40	12.30	11.40	5.00	12.30
	cov. effect 2	0.60	0.10	0.60	-1.00	0.10
Logistic	Intercept	-1.10	0.50	-1.10	0.50	-1.00
	cov. effect	0.60	-0.20	0.60	-0.20	0.40
Survival	Intercept	2.00	2.50	2.00	2.20	2.50
	scale	1.70	1.98	1.70	1.80	1.98
	cov. effect 1	3.00	3.90	3.00	3.40	3.90
	cov. effect 2	4.20	4.90	4.20	4.70	4.90

Table 2: True parameter values in the regression simulation scenarios. A linear, logistic and survival models are studied in the one and two breakpoints settings. The linear model is homoscedastic with an error standard deviation equal to 2.5 in the two settings. In the one breakpoint setting, the breakpoints are equal to 553, 112 and 666 in the linear, logistic and survival models, respectively. In the two breakpoint setting, the breakpoints are equal to 333 and 666 in the linear and logistic models, they are equal to 375 and 689 in the survival model.

5.3 Implementation of the breakpoint tests in regression models

In this section, we consider the statistical test developed in Section 4 for the one breakpoint situation. This test is based on a permutation implementation where Theorems 1 and 2 are used for the computation of the distribution of the statistical test under H_0. The idea is simple: we randomly shuffle the order of the data $B = 1,000$ times, and we consider that each shuffled sample is a realization of the test statistic. This realization is calculated using the approximations developed in Theorems 1 and 2 and therefore the max-EM algorithm does not need to be run. In practice, once this step has been performed, the p-value of the test can be computed by simply comparing the observed value of the statistical test on the original sample (using again Theorems 1 and 2) with the distribution of the statistical test under H_0 obtained with the permutation implementation. By construction, the statistical test is automatically well calibrated under H_0: the rejection rate of the α level test under H_0 is equal to α. However, it is of interest to investigate the power of the statistical test under various alternatives. This simulation experiment is conducted under the three regression models introduced in Section 5.2. In the linear model, Theorem 1 is used for samples larger than 100, that is for $n_1 = 101, \ldots, 900$, in combination with Theorem 2 which is used for small samples (that is for $n_1 < 100$ and

$n = 1,000$		Linear Model bp = 553	Logistic Model bp = 112	Survival Model bp = 666
One bp	MSE($\hat{\boldsymbol{\theta}}$)	0.86471	1.41481	0.10759
	BIAS^2($\hat{\boldsymbol{\theta}}$)	0.00280	0.01496	0.00157
	VAR($\hat{\boldsymbol{\theta}}$)	0.86191	1.39566	0.10602
	MAPE($\hat{\boldsymbol{\theta}}$)	4.37378	2.74543	0.26998
	ACCE(bp)	0.01367	0.01011	0.00160
		bp = $(333, 666)$	bp = $(333, 666)$	bp = $(375, 689)$
Two bp	MSE($\hat{\boldsymbol{\theta}}$)	1.74872	1.27661	0.26253
	BIAS^2($\hat{\boldsymbol{\theta}}$)	0.00435	0.00473	0.00220
	VAR($\hat{\boldsymbol{\theta}}$)	1.74437	1.27188	0.26033
	MAPE($\hat{\boldsymbol{\theta}}$)	5.46586	2.38020	0.52188
	ACCE(bp)	0.00221	0.01779	0.01122

Table 3: Results for the max-EM algorithm with Binary Segmentation (BS) initialization in one and two breakpoint regression models. The first model is a linear homoscedastic regression model with two covariates, the second model is a logistic model with intercept and one covariate and the third model is a Weibull survival regression model with two covariates. The Mean Squared Error (MSE) of the estimated parameters, decomposed as the variance (VAR) plus squared bias (BIAS^2), along with the Mean Absolute Percentage Error (MAPE) of the estimated parameters and the ACCE of the estimated breakpoints are provided. The values of the true parameters can be found in Table 2.

$n_1 \geq 900$). In the logistic and survival models, only Theorem 1 is used since the properties of the corresponding estimators are solely asymptotic. This amounts to constraining our test to detect a breakpoint for $n_1 \geq 100$ and $n_1 < 900$ only. We start by considering the same parameter values as before (first scenario) and we then increase the difficulty in the segmentation detection in the second and third scenarios. The description of those scenarios with the corresponding values of the regression parameter values are given in Table 4.

First, the log-likelihood ratio is computed on a single sample, for all possible breakpoint values and for all three models, in the first scenario. The value of the likelihood ratios with respect to the breakpoint values are displayed in Figure 2. On these samples, we clearly see that the maximum of the log-likelihood ratio is very close to the true value which is represented in dotted vertical lines in the figure. Then, the histograms of the statistical test are displayed in Figure 3 in all situations, based on $M = 1,000$ Monte-Carlo replications. The more the distribution under H_1 is far from the distribution under H_0, the more powerful the test is. For reference, the empirical 0.95 quantile of the distribution under H_0 is shown as a vertical dotted line in order to visualize the power of the test for a 5% level test. We clearly see that the power of the tests decreases as the distribution of the test statistic between H_0 and H_1 gets more similar (from left to right). For the linear model, the rejection rate under a 5% level test is equal to 1, 0.903, 0.577 for the left, middle and right panels, respectively. For the logistic model, the rejection rate under a 5% level test is equal to 0.998, 0.851, 0.558 for the left, middle and right panels, respectively. For the survival model, the rejection rate under a 5% level test is equal to 1, 0.928, 0.601 for the left, middle and right panels, respectively. Of importance, the permutation method is extremely fast to implement due to our approximations in Theorems 1 and 2. For illustration, the computation of the $M = 1,000$ samples used to derive the empirical distribution of the statistical test under H_0 is achieved in 18 seconds on average, over all three scenarios, on a typical personal computer with 32Go of RAM.

		First scenario		Second scenario		Third scenario	
		θ_1^*	θ_2^*	θ_1^*	θ_2^*	θ_1^*	θ_2^*
Linear	Intercept	1.00	2.00	1.00	1.00	1.00	1.00
($\sigma = 2.5$)	cov. effect 1	11.40	12.30	11.00	12.30	11.40	12.30
	cov. effect 2	0.60	0.10	0.10	0.10	0.10	0.10
Logistic	Intercept	-1.10	0.50	0.50	0.50	0.50	0.50
	cov. effect	0.60	-0.20	1.20	-0.20	0.80	-0.20
Survival	Intercept	2.00	2.50	2.00	2.00	2.00	2.00
	scale	1.70	1.98	1.70	1.70	1.70	1.70
	cov. effect 1	3.00	3.90	3.10	3.90	3.30	3.90
	cov. effect 2	4.20	4.90	4.90	4.90	4.90	4.90

Table 4: Parameter values in the regression simulation scenarios for the statistical tests. A linear, logistic and survival models are studied in the one breakpoint setting. The linear model is homoscedastic with an error standard deviation equal to 2.5 in all three scenarios. The breakpoints are equal to 553, 112 and 666 in the linear, logistic and survival models, respectively.

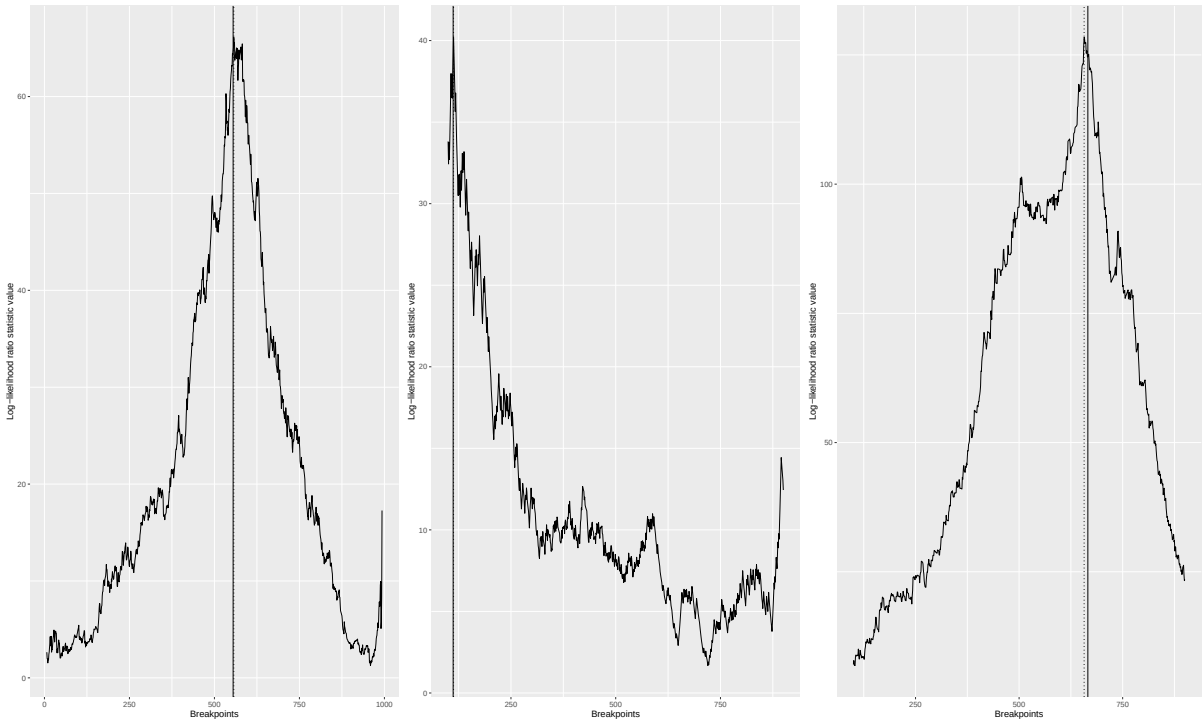

Figure 2: Log-likelihood ratio statistic for the test introduced in Section 4, in the linear (left panel), logistic (middle panel) and Weibull Cox (right panel) models. In each model, the true breakpoint is displayed as a plain vertical line and is equal to 555, 112 and 666, respectively. The maximum of the log-likelihood ratio statistic is displayed as a dotted vertical line. The log-likelihood ratio statistic test is computed on a single sample using the approximations derived in Theorems 1 and 2.

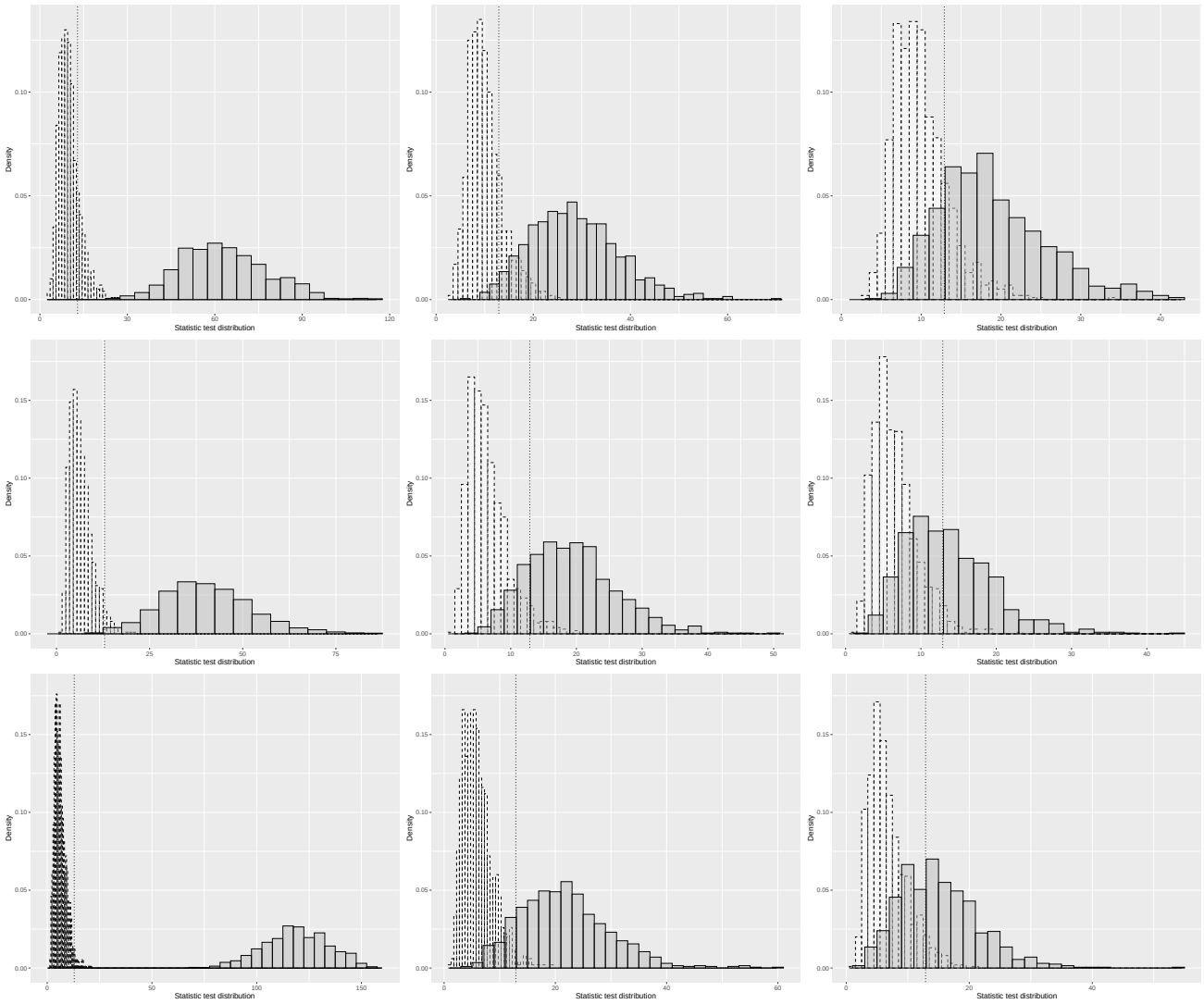

Figure 3: Distribution of the test statistics developed in Section 4, in the linear (top row), logistic (middle row) and survival (bottom row) models. The distribution under (H_0) is represented by a white histogram with dotted contour line while the distribution under (H_1) is represented by grey histogram with plain contour line. The 0.95 quantile under (H_0) is shown as a vertical dotted line. The three columns correspond to three scenarios with decreasing power (from left to right). Those scenarios are described in details in Section 5.3 and Table 4. The log-likelihood ratio statistic test is computed using the approximations derived in Theorems 1 and 2.

6 Applications

6.1 Tendency breakpoint detection on the bike sharing dataset

In this section we study the bike sharing dataset, available online on the UCI website. This dataset comprises the daily counts of the number of total rental bikes in a city from January 1, 2011 until December 31, 2012. It contains a total of 731 values (365 in 2011 and 366 in 2012). The time series is displayed in Figure 4. The aim is to study the trend of this time series and to detect change of trends with respect to the date. For that purpose, we use a simple linear regression model with intercept and the date as the only covariate. In the breakpoint analysis, we assume the model is homoscedastic, that is the variance of the residuals is the same in all segments. We start by performing the one breakpoint test. Using Theorem 1, we compute the

test statistic on those data and we simulate the test statistic under H_0 based on $B = 1,000$ random permutations of the data. The results are shown in Figure 5. The log-likelihood ratio statistic computed on the data is displayed on the left panel. We observe that the maximum is attained in 23 September 2012 and equals 286.42. The empirical distribution under H_0 is displayed on the right panel with the 0.95 empirical quantile represented as a vertical dotted line. We clearly see that, under H_0, the test statistic takes much lower values than 286.42 and therefore the test is extremely significant with a p-value equal to 0. When looking at the log-likelihood ratio statistic (on the left panel) we observe many other local maximums which have a value quite large as compared to the values taken by the test statistic under H_0. This suggests that the data may contain more breakpoints.

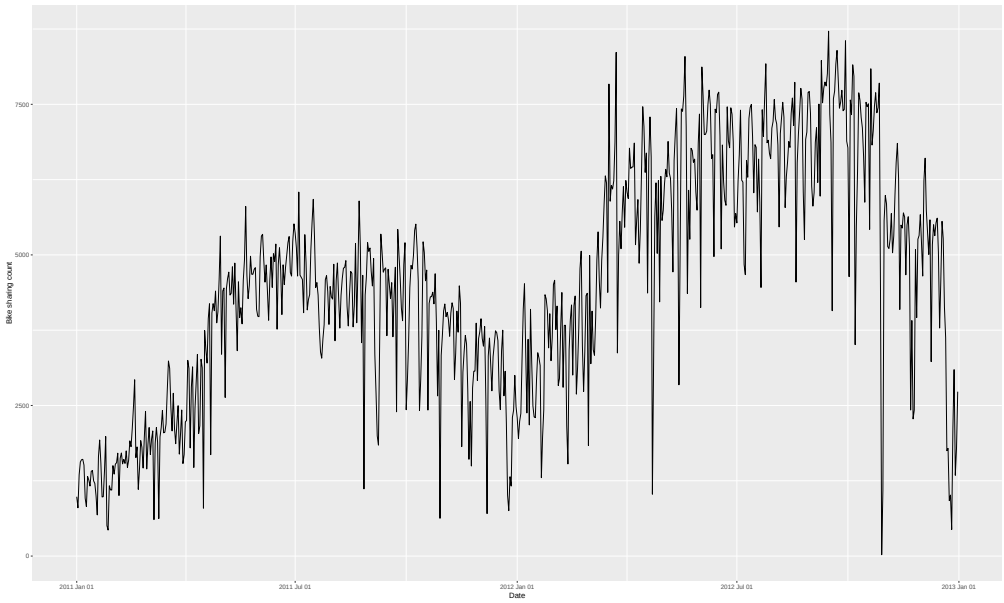

Figure 4: Time series of bike sharing counts. The data are reported daily, from January 1, 2011 until December 31, 2012.

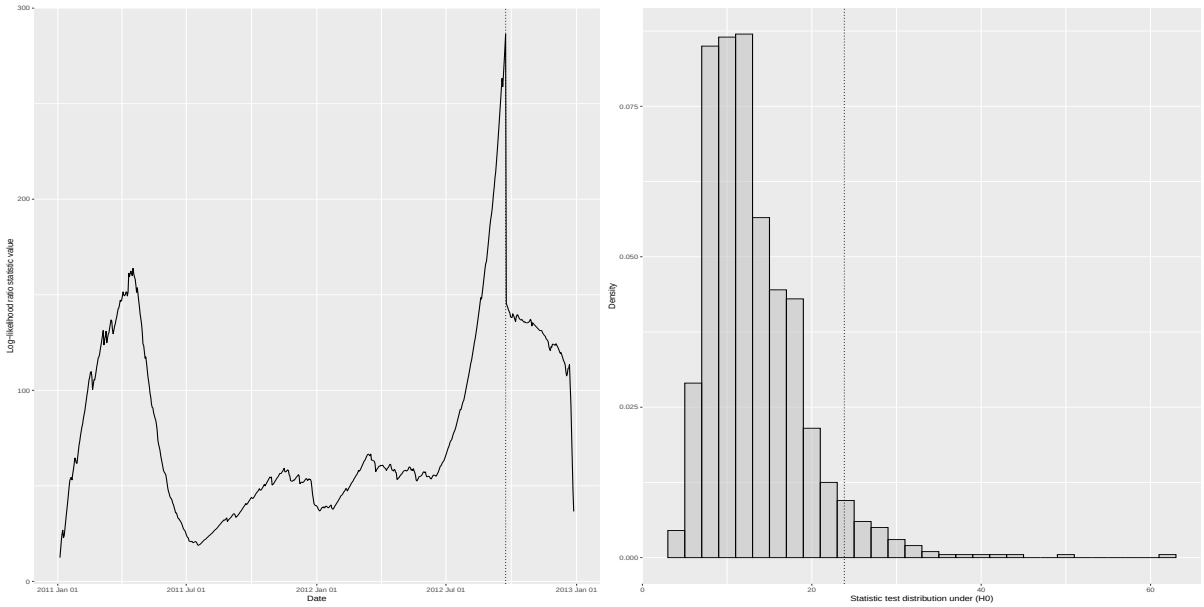

Figure 5: Statistical test for the one breakpoint detection problem for the bike sharing counts dataset. Left panel: log-likelihood ratio statistic computed on the original data. The maximum of the statistic is displayed as a vertical dotted line and equals 286.42. Right panel: distribution of the test statistic under H_0. The distribution is obtained from $1,000$ permutations of the data. The 0.95 quantile is shown as a dotted line. For both plots, the approximated expression of the log-likelihood ratio statistic was derived from Theorem 1. The value of the test statistic obtained in the left panel (286.42) corresponds to a p-value equal to 0.

We then apply our max-EM algorithm to the data, with a number of breakpoints ranging from 1 to 6. In Table 5, we present the results of the different analyzes with the values of the estimated slopes and the value of the BIC computed using the expression introduced in Section 3.5. The values of the intercepts along with the dates at which the breakpoints occur can be found in Supplementary Materials. The plots of the linear models derived from these estimated parameters is also displayed in Figure 6. Up to five breakpoints, as the number of breakpoints increases, we clearly see an improvement in the data fitting, with very different values of slopes in two consecutive segments. On the contrary, in the six breakpoints model, the third and fourth breakpoints occur over a short period of time (2011-11-15 and 2011-12-22) with a change of slope sign (-3.85 and 12.73) that does not seem to fit the data. Looking at the BIC value, it turns out that the five segments model is preferred over the six breakpoints model which is in agreement with Table 5 and Figure 6.

bp	Slope values							BIC
0	5.7688							12791.2900
1	7.7393	-35.5764						12599.4121
2	12.5053	14.3050	-35.5764					12411.8734
3	16.3069	-5.6481	7.1842	-35.5764				12193.2309
4	16.3069	-3.2393	10.7407	6.7402	-35.5764			12154.7065
5	14.2500	13.6033	-8.9810	26.3382	6.6382	-35.5764		**12149.2600**
6	14.2500	13.6033	-3.8540	12.7326	26.3382	6.6382	-35.5764	12150.7580

Table 5: Estimated slope values obtained from the max-EM algorithm with the bike sharing counts dataset. The daily counts of shared bikes is modeled using a piecewise linear regression with respect to the dates in different models ranging from 0 to 6 breakpoints. The BIC is also reported in the last column.

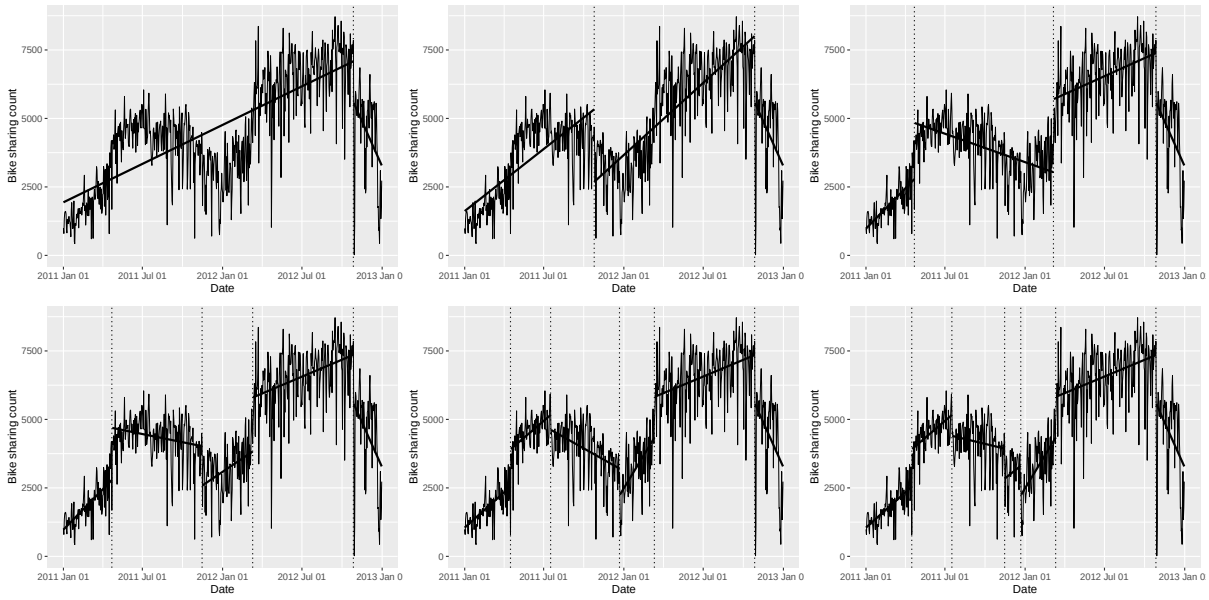

Figure 6: Tendency breakpoint detection and piecewise linear models implemented on the bike sharing dataset with various number of breakpoints, ranging from 1 to 6. The minimal value of the BIC is obtained for the 5 breakpoint model.

6.2 Heterogeneity of the effect of fasting blood sugar on heart disease

In this second real data application we study the heart disease dataset available on the UCI website. On this dataset of size $n = 303$, the goal is to detect an heterogeneity in the effect of fasting blood sugar (fbs) on the risk of developing a heart disease (54.46% of the patients have a diagnostic of heart disease). In order to do so, we use the following continuous covariates: age, resting blood pressure on admission to the hospital (trestbps, in mm per Hg), cholesterol (chol, in mg per dl), maximum heart rate achieved (thalach) and ST depression induced by exercise relative to rest (oldpeak). Those covariates are used to construct a "proximity space" which allows us to order the individuals. Then we apply the max-EM algorithm for the logistic regression model where the outcome variable is the diagnostic of heart disease (1 yes, 0 no) and the only covariate is fbs. This covariate is binary, with value 1 when the fasting blood sugar exceeds 120 mg/dl and 0 when it is below this threshold. The idea behind the construction of the proximity space is to find an order of individuals where two individuals whose ranks are close (respectively, far) to each other should be similar (respectively, different) in terms of covariates. To do so, we fit a principal curve [see 26] and we project the individuals on this curve. This is done using the `principal_curve` function from the `princurve` R package. When the principal curve algorithm has converged, the order of individuals is obtained from the location on the curve (which is a space of dimension 1) and we apply the max-EM algorithm to detect possible breakpoints.

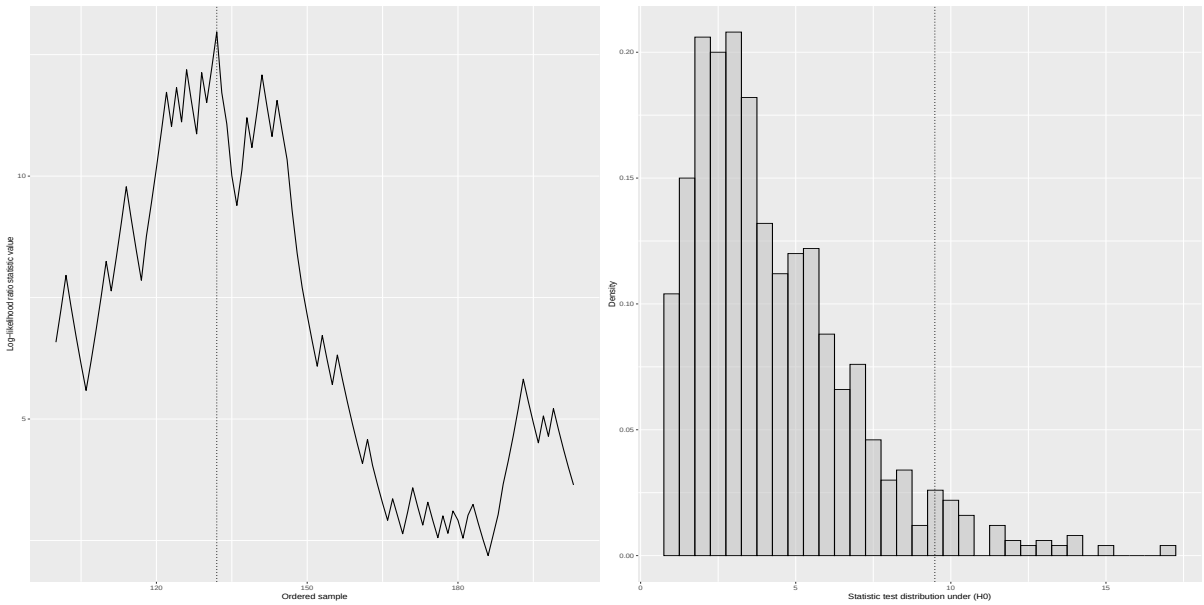

Figure 7: Statistical test for the one breakpoint detection problem for the heart disease dataset. Left panel: log-likelihood ratio statistic computed on the data ordered from the proximity space (this space was constructed from a principal curve fit on the covariates) based on a logistic regression model. The maximum of the statistic is displayed as a vertical dotted line and equals 12.96. Right panel: distribution of the test statistic under H_0. The distribution is obtained from $1,000$ permutations of the data. The 0.95 quantile is shown as a dotted line. For both plots, the approximated expression of the log-likelihood ratio statistic was derived from Theorem 1. The value of the test statistic obtained in the left panel (12.96) corresponds to a p-value equal to 0.011.

Before implementing the max-EM algorithm, we start by the test statistic for the one breakpoint scenario. In Figure 7, the log-likelihood ratio statistic is displayed on the left-panel for the ordered data with the maximum attained at the value 12.96. On the right-panel, the distribution of the test statistic under H_0 is obtained based on Theorem 1 with the 0.95 empirical quantile (equal to 9.48) represented as a vertical dotted line. The p-value is simply the probability that this density is greater than 12.96 and it equals 0.011. The test is therefore highly significant and suggests that the effect of fbs is heterogeneous according to a breakpoint on the principal curve space. Since the ordering of individuals on this space was obtained based on covariates proximity, this suggests an interaction effect of covariates/fbs on the diagnosis of heart disease. Next, the max-EM algorithm is implemented with different breakpoint models. The result of the BIC along with the odds ratios for fbs on the diagnosis of heart disease are displayed in Table 6. We observe that the model with minimum value for the BIC is the one breakpoint model for which the odds ratios in the two segments are equal to 0.56 and 1.12, respectively. This means that fbs has a strong protective effect for individuals in segment 1 and a slightly worsening effect for individuals in segment 2. In the one breakpoint model, the two segments are of size $n = 132$ and $n = 171$, respectively.

In order to investigate what can cause the odds ratios to be twice as big in segment 2 as compared to segment 1, we have also compared the distributions of the covariates in the two segments. We present, in Figure 8, the univariate distributions of the 5 covariates. Since the oldpeak variable has a lot of zeros (which means the patient had no ST depression), the distribution of this variable is for the positive values only (for reference, there are a total of 99 individuals with a value of oldpeak equal to 0 in both segments, which correspond to 30.3% and 34.5% of oldpeak values equal to 0 in segments 1 and 2, respectively). We observe that the main covariate that distinguishes the two segments is cholesterol with much lower values in segment 2 as compared to segment 1 (median with interquartile range equals $282\,[264, 307]$

bp	Odds ratios for fbs			BIC
0	0.8540			428.8278
1	0.5611	1.1209		**427.1403**
2	0.5611	0.9698	4.5000	432.5396

Table 6: Estimated odds ratios obtained from the max-EM algorithm with the heart disease dataset. The odds for fasting blood sugar on diagnosis of heart disease is modelled using logistic regression based on a proximity space constructed from the principal curve of 5 different covariates. Models ranging from 0 to 2 breakpoints are presented. The BIC is also reported in the last column.

and $214\,[197, 232]$ in segments 1 and 2, respectively). Then, individuals in segment 2 tend to be younger ($57\,[51, 63]$ in segment 1 and $53\,[44, 59]$ in segment 2), with a higher value of thalach ($148.5\,[130, 162]$ in segment 1 and $157\,[140.5, 170]$ in segment 2). Regarding the oldpeak variable, there are more patients with no ST depression in segment 2, but among those who had ST depression, the ST depression value tends to bee slightly larger in segment 2 than in segment 1 ($1.25\,[0.8, 2]$ in segment 1 and $1.4\,[0.6, 2.02]$ in segment 2). The correlation between all pair of variables was also studied and compared between each segment. We present all the pairwise correlation values in Table 4 in Supplementary Material along with the scatter plots of some of the variables in Figure 1. Focusing on only the strongest associations between pair of variables, we see that: age and thalach are negatively correlated with a correlation equal to -0.253 and -0.461 in segments 1 and 2, respectively; age is positively correlated with trestbps (it is equal to 0.209 and 0.301 in segments 1 and 2, respectively); thalach is positively correlated with cholesterol (it is equal to 0.187 and 0.228 in segments 1 and 2, respectively) and oldpeak is positively correlated with trestbps (it is equal to 0.253 and 0.151 in segments 1 and 2, respectively).

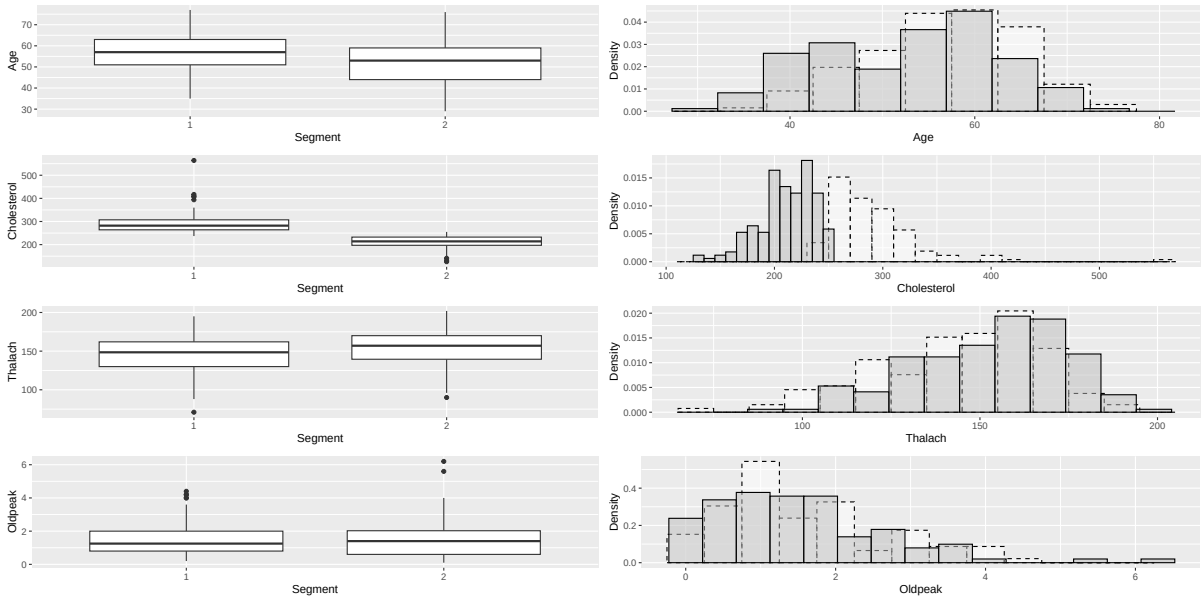

Figure 8: Univariate distributions of the covariates used to construct the proximity space between the two segments. The breakpoint, and therefore the two segments, were obtained from the max-EM algorithm. Left panel: boxplots of the covariates. Right panel: histograms of the covariates in white with dotted contour lines and in grey with plain contour line for the data in segment 1 and segment 2, respectively.

7 Discussion and perspectives

In this work we presented a new method for breakpoint detection in regression modeling. Our method, called max-EM, which combines the CEM algorithm with HMM, is an extension of previous approaches on the topic based on the standard EM algorithm. We showed that it is tailored to the breakpoint detection problem: when the targeted likelihood is a function of both the regression parameters and breakpoint locations, we proved that each iteration of the max-EM algorithm increases this likelihood. We also presented two strategies for the initialization of the algorithm and we proposed to use the standard BIC in practice to find the correct number of breakpoints. Finally, a new statistical test for the one breakpoint situation based on the likelihood ratio for all possible segments has been studied: we established an asymptotic approximation that allows to compute this test in an efficient and fast way.

As compared to the GFPOP algorithm, our method does not provide an exhaustive exploration of all possible segmentations but rather, is based on statistical models and aims at increasing the likelihood at each iteration. Using our initialization methods, our method becomes more stable and seems to be able to reach global maximums. It is extremely fast, even though the initialization step requires to run the algorithm several times. Our simulation experiments tend to favor the BS initialization over the FL initialization, in terms of computation time - accuracy balance. Importantly, our method can consider a very wide variety of regression models, a feature that is not possible using dynamic programming. In our simulation settings and in the analyzes of real data, we indeed considered linear, logistic, Poisson, and AFT regression models. We showed that in all these regression settings, with a number of breakpoints ranging from 1 to 5, our method was extremely performant, both in terms of breakpoint detection and parameters estimation. The statistical test was also studied under various regression models. It showed a correct rejection rate under the null hypothesis and a strong power under some alternative hypothesis. This was illustrated on the two studied datasets where the test was highly significant, in particular for the bike sharing dataset. Using the BIC to choose the correct number of breakpoints seemed also to be a powerful approach. In the two datasets we found relevant segments where the distribution of the data was clearly different between segments. Those applications showed the versatility of our approach. In the bike sharing dataset, it was used to detect change of trends in the number of total rental bikes with respect to the date. In the heart disease data, it was combined with the construction of principal curves to construct a proximity space on the covariates. This proximity space was then used to define the order of individuals and combined with the max-EM algorithm this enabled us to detect different effects of fasting blood sugar on the occurrence of heart disease. The segments were composed of covariates with similar values among segments and different values between segments. Analyzing the distribution of the covariates inside the two segments, the whole procedure enabled us to detect complex interactions between the effect of fasting blood sugar and the other covariates on the occurrence of heart disease.

A setting that we did not investigate in this work occurs when some regression parameters are imposed to be shared between segments. This is an attractive modeling approach, however the current method does not support this feature. This is due to the fact that the parameter update step of the algorithm simply consists in estimating the parameters in each segment (thus leading to different estimations per segments). In our simulations, we considered scenarios where some of the parameters are common over the segments: we simulated data following the homoscedastic linear regression and the homoscedatic AFT model. In those settings, our method did not take advantage of the homoscedastic structure of the data but still provided accurate parameter estimations. However, it would be of interest to develop a method that explicitly incorporates this feature in the estimation method. In particular, this would be extremely relevant in the context of censored data, where one wants to use the popular Cox model. When the variable of interest is a time variable, it might be relevant to detect changes in terms of hazard ratios

of a covariate of interest between segments and to keep the baseline common to all segments. This modeling option would need further work, both for the max-EM algorithm and for the one breakpoint statistical test. Regarding the test, this would be particularly relevant as our current approach might detect heterogeneity due to baseline differences among segments, when one might only be interested in changes in the covariate effect. This is left to future research work.

Acknowledgement

The authors warmly thank Guillem Rigaill and Vincent Runge for our fruitful discussions on the GFPOP algorithm. This work is part of the project entitled "A new method for the detection of gene-environment interactions in cancer studies" and was funded by the Ligue Nationale Contre le Cancer (LNCC).

A Appendix

A.1 Proof of Proposition 1

At the $(m+1)$th step, we have for $k = R_i^{max(m+1)}$, for all $k' = 1, \dots, K$, for all $i = 1, \dots, n$, $F_i^{\max}(k; \boldsymbol{\theta}^{(m)}) B_i^{\max}(k; \boldsymbol{\theta}^{(m)}) \geq F_i^{\max}(k'; \boldsymbol{\theta}^{(m)}) B_i^{\max}(k'; \boldsymbol{\theta}^{(m)})$. From Equation (5) and the definition of ℓ_n in Equation (1), we therefore have

$$\ell_n\left(\boldsymbol{\theta}^{(m)}; n_{1:(K-1)}^{(m+1)}\right) \geq \ell_n\left(\boldsymbol{\theta}^{(m)}; n_{1:(K-1)}^{(m)}\right).$$

Now, from the M-step, $\boldsymbol{\theta}^{(m+1)}$ is the maximizer of $\ell_n\left(\boldsymbol{\theta}; n_{1:(K-1)}^{(m+1)}\right)$ and consequently

$$\ell_n\left(\boldsymbol{\theta}^{(m+1)}; n_{1:(K-1)}^{(m+1)}\right) \geq \ell_n\left(\boldsymbol{\theta}^{(m)}; n_{1:(K-1)}^{(m+1)}\right).$$

This proves that the sequence $\left(\ell_n\left(\boldsymbol{\theta}^{(m)}; n_{1:(K-1)}^{(m)}\right)\right)_{m\geq 1}$ is increasing. Since there is a finite number of partition of the segments $R_{1:n}$ under the contraint $R_n = K$ and since $e_i(k; \theta_k)$ is bounded, the log-likelihood $\ell_n\left(\boldsymbol{\theta}^{(m)}; n_{1:(K-1)}^{(m)}\right)$ converges towards a finite value. Moreover the maximum is unique by assumption and as a consequence $\left(\boldsymbol{\theta}^{(m)}; n_{1:(K-1)}^{(m)}\right)_{m\geq 1}$ converges towards a stationary point.

A.2 EM and max-EM algorithms

A.2.1 MAP in the E-step of the max-EM algorithm

The $F_i^{\max}(k;\theta)$ and $B_i^{\max}(k;\theta)$ can be combine to compute

$$\begin{aligned} F_i^{\max}(k;\theta) B_i^{\max}(k;\theta) &\equiv \underbrace{F_i^{\max}(R_i = k;\theta) B_i^{\max}(R_i = k;\theta)}_{=\text{MAP in } R_i}. \\ &= \max_{R_{1:i-1}, R_{(i+1):(n-1)}} \mathbb{P}(R_1, \dots, R_{i-1}, R_i = k, R_{i+1}, \dots, R_n = K, X_{1:n}|\theta) \end{aligned}$$

Proof: From

$$F_i^{\max}(k;\theta) = \max_{R_1,\dots,R_{i-1}} \mathbb{P}(R_{1:i-1}, R_i = k, X_{1:i}|\theta),$$

and

$$B_i^{\max}(k;\theta) = \max_{R_{i+1},\dots,R_{n-1}} \mathbb{P}(R_{(i+1):(n-1)}, R_n = K, X_{(i+1):n}|R_i = k;\theta).$$

We compute the product $F_i^{\max}(k;\theta) \times B_i^{\max}(k;\theta)$ as

$$F_i^{\max}(k;\theta) \times B_i^{\max}(k;\theta) = \max_{R_1,\dots,R_{i-1}} \mathbb{P}(R_{1:i-1}, R_i = k, X_{1:i}) \times \max_{R_{i+1},\dots,R_{n-1}} \mathbb{P}(R_{(i+1):(n-1)}, R_n = K, X_{(i+1):n} | R_i = k;\theta)$$

Then, considering that

$$\begin{aligned}&\mathbb{P}(R_{(i+1):(n-1)}, R_n = K, X_{(i+1):n} | R_i = k;\theta)\\ &= \mathbb{P}(R_{(i+1):(n-1)}, R_n = K, X_{(i+1):n} | R_i = k, R_{1:(i-1)}, X_{1:i};\theta)\end{aligned}$$

we obtain

$$\begin{aligned}&\mathbb{P}(R_{1:i-1}, R_i = k, X_{1:i};\theta)\\ &\times \mathbb{P}(R_{(i+1):(n-1)}, R_n = K, X_{(i+1):n} | R_i = k, R_{1:(i-1)}, X_{1:i};\theta)\\ &= \mathbb{P}(R_{(i+1):(n-1)}, R_n = K, X_{(i+1):n}, R_i = k, X_{1:i}, R_{1:(i-1)} | \theta)\end{aligned}$$

Thus

$$F_i^{\max}(k;\theta) B_i^{\max}(k;\theta) = \max_{R_{1:(i-1)}, R_{(i+1):(n-1)}} \mathbb{P}(R_{(i+1):(n-1)}, R_n = K, X_{1:n}, R_i = k, R_{1:(i-1)} | \theta)$$

A.3 Forward Backward and Max-Forward Max-Backward algorithms

A.3.1 Forward Backward algorithm in logarithmic scale

In order to avoid the underflow problem, we factor the results into a logarithmic scale:

$$e_i(k;\theta) = \mathrm{e}^{\ell_i} \tilde{e}_i(k;\theta), \qquad F_i(k;\theta) = \mathrm{e}^{L_i} \tilde{F}_i(k;\theta) \quad \text{and} \quad B_i(k;\theta) = \mathrm{e}^{M_i} \tilde{B}_i(k;\theta),$$

with

$$\mathrm{e}^{\ell_i} = \max_k e_i(k;\theta), \qquad \mathrm{e}^{L_i} = \max_k F_i(k;\theta) \quad \text{and} \quad \mathrm{e}^{M_i} = \max_k B_i(k;\theta),$$

From there, we find for the forward quantities (the same holds for the backward quantities):

$$F_i(k;\theta) = \sum_j F_{i-1}(j;\theta)\pi(j,k)e_i(k;\theta) \;\Leftrightarrow\; \mathrm{e}^{L_i} \tilde{F}_i(k;\theta) = \mathrm{e}^{L_{i-1}+\ell_i} \sum_j \tilde{F}_{i-1}(j;\theta)\pi(j,k)\tilde{e}_i(k;\theta)$$

Therefore,

$$L_i = L_{i-1} + \ell_i + \max_k \log\Big(\sum_j \tilde{F}_{i-1}(j;\theta)\pi(j,k)\tilde{e}_i(k;\theta)\Big) \quad \text{and } \tilde{F}_i(k;\theta) \propto \sum_j \tilde{F}_{i-1}(j;\theta)\pi(j,k)\tilde{e}_i(k;\theta)$$

and, in the same way:

$$M_{i-1} = M_i + \ell_i + \max_j \log\Big(\sum_k \pi(j,k)\tilde{e}_i(k;\theta)\tilde{B}_i(k;\theta)\Big) \quad \text{and } \tilde{B}_{i-1}(j;\theta) \propto \sum_k \pi(j,k)\tilde{e}_i(k;\theta)\tilde{B}_i(k;\theta)$$

A.3.2 Max-Forward Max-Backward algorithm in logarithmic scale

similarly with the forward backward algorithm, we find for the forward quantities (the same holds for the backward quantities):

$$F_i^{\max}(k;\theta) = \max_j F_{i-1}^{\max}(j;\theta)\pi(j,k)e_i(k;\theta) \;\Leftrightarrow\; \mathrm{e}^{L_i} \tilde{F}_i^{\max}(k;\theta) = \mathrm{e}^{L_{i-1}+\ell_i} \max_j \tilde{F}_{i-1}^{\max}(j)\pi(j,k)\tilde{e}_i(k;\theta)$$

Therefore,

$$L_i = L_{i-1} + \ell_i + \max_k \log\Big(\max_j \tilde{F}_{i-1}^{\max}(j)\pi(j,k)\tilde{e}_i(k;\theta)\Big) \text{ and } \tilde{F}_i^{\max}(k;\theta) \propto \max_j \tilde{F}_{i-1}^{\max}(j)\pi(j,k)\tilde{e}_i(k;\theta)$$

and, in the same way:

$$M_{i-1} = M_i + \ell_i + \max_j \log\Big(\max_k \pi(j,k)\tilde{e}_i(k;\theta)\tilde{B}_i^{\max}(k;\theta)\Big) \text{ and } \tilde{B}_{i-1}^{\max}(j) \propto \max_k \pi(j,k)\tilde{e}_i(k;\theta)\tilde{B}_i^{\max}(k;\theta)$$

A.4 Statistical Tests: Theorem Proofs

A.4.1 Proof of Theorem 1

We first recall that

$$\ell_n(\theta_1,\theta_2) = \sum_{i=1}^{n_1} \log\left(\mathbb{P}(X_i;\theta_1)\right) + \sum_{i=n_1+1}^{n} \log\left(\mathbb{P}(X_i;\theta_2)\right),$$
$$\tilde{\ell}_n(\theta) = \sum_{i=1}^{n} \log\left(\mathbb{P}(X_i;\theta)\right) = \sum_{i=1}^{n_1} \log\left(\mathbb{P}(X_i;\theta)\right) + \sum_{i=n_1+1}^{n} \log\left(\mathbb{P}(X_i;\theta)\right), \tag{A.7}$$

and $(\hat{\theta}_1,\hat{\theta}_2) = \arg\max_{\theta_1,\theta_2} \ell_n(\theta_1,\theta_2)$, $\hat{\theta}_0 = \arg\max_\theta \tilde{\ell}_n(\theta)$. It is clear that $\tilde{\ell}_n(\theta) = \ell_n(\theta,\theta)$ but it should be noted that the gradient and Hessian matrix for $\tilde{\ell}_n$ and ℓ_n are different even if they are evaluated at the same parameter value. In particular, $\nabla\ell_n(\theta_1,\theta_2)$ is a $2d$ dimensional vector where the first d components contain the vector $\sum_{i=1}^{n_1} \nabla \log\left(\mathbb{P}(X_i;\theta_1)\right)$ and the last d components contain the vector $\sum_{i=n_1+1}^{n} \nabla \log\left(\mathbb{P}(X_i;\theta_2)\right)$. The Hessian matrix $\nabla^2\ell_n(\theta_1,\theta_2)$ is a $2d \times 2d$ matrix which can be decomposed as four $d \times d$ block matrices in the following way:

$$\nabla^2\ell_n(\theta_1,\theta_2) = \begin{pmatrix} \sum_{i=1}^{n_1} \nabla^2 \log\left(\mathbb{P}(X_i;\theta_1)\right) & 0_{d\times d} \\ 0_{d\times d} & \sum_{i=n_1+1}^{n} \nabla^2 \log\left(\mathbb{P}(X_i;\theta_2)\right) \end{pmatrix}$$

Note first that under general maximum likelihood theory, $\hat{\theta}_1,\hat{\theta}_2$ and $\hat{\theta}_1,\hat{\theta}_0$ all converge to θ_0^* under (H_0), when $n_1 \to \infty$ and $n - n_1 \to \infty$. From Taylor developments and using the fact that $\nabla\ell_n(\hat{\theta}_0) = 0$ we have:

$$\tilde{\ell}_n(\theta_0^*) = \tilde{\ell}_n(\hat{\theta}_0) + \frac{1}{2}(\theta_0^* - \hat{\theta}_0)^\top \nabla^2\ell_n(\theta_0')(\theta_0^* - \hat{\theta}_0),$$
$$\hat{\theta}_0 - \theta_0^* = \left(-\nabla^2\ell_n(\theta_0'')\right)^{-1} \nabla\ell_n(\theta_0^*), \tag{A.8}$$

where θ_0' and θ_0'' are on the real line between θ_0^* and $\hat{\theta}_0$. From the law of large numbers and the consistency of $\hat{\theta}_0$ we have that, under (H_0), $-\nabla^2\ell_n(\theta_0')/n$, $-\nabla^2\ell_n(\theta_0'')/n$ converge towards the Fisher information

$$I(\theta_0^*) = -\mathbb{E}[\nabla^2 \log\left(\mathbb{P}(X_i;\theta_0^*)\right)].$$

From the central limit theorem we have that, under (H_0),$\nabla\ell_n(\theta_0^*)/\sqrt{n}$ converges toward a centered Gaussian variable in distribution. Using Slutsky's theorem, we directly obtain

$$\tilde{\ell}_n(\hat{\theta}_0) = \tilde{\ell}_n(\theta_0^*) + \frac{1}{2}\nabla\tilde{\ell}_n(\theta_0^*)^\top \left(-\nabla^2\tilde{\ell}_n(\theta_0^*)\right)^{-1} \nabla\tilde{\ell}_n(\theta_0^*) + o_{\mathbb{P}}(1). \tag{A.9}$$

Then, using the decomposition in the right-hand side of Equation (A.7) for $\nabla\tilde{\ell}_n(\theta_0^*)$, we have:

$$\begin{aligned}
\tilde{\ell}_n(\hat{\theta}_0) = \tilde{\ell}_n(\theta_0^*) & \\
&+ \frac{1}{2}\left[\left(-\frac{1}{n}\sum_{i=1}^{n} \nabla^2 \log\left(\mathbb{P}(X_i;\theta_0^*)\right)\right)^{-1/2} \frac{1}{\sqrt{n}} \sum_{i=1}^{n_1} \nabla \log\left(\mathbb{P}(X_i;\theta_0^*)\right)\right]^{\otimes 2} \\
&+ \frac{1}{2}\left[\left(-\frac{1}{n}\sum_{i=1}^{n} \nabla^2 \log\left(\mathbb{P}(X_i;\theta_0^*)\right)\right)^{-1/2} \frac{1}{\sqrt{n}} \sum_{i=n_1+1}^{n} \nabla \log\left(\mathbb{P}(X_i;\theta_0^*)\right)\right]^{\otimes 2} \\
&+ \frac{1}{\sqrt{n}}\sum_{i=1}^{n_1} \nabla \log\left(\mathbb{P}(X_i;\theta_0^*)\right)^\top \left(-\frac{1}{n}\sum_{i=1}^{n} \nabla^2 \log\left(\mathbb{P}(X_i;\theta_0^*)\right)\right)^{-1} \frac{1}{\sqrt{n}} \sum_{i=n_1+1}^{n} \nabla \log\left(\mathbb{P}(X_i;\theta_0^*)\right) \\
&+ o_{\mathbb{P}}(1).
\end{aligned}$$

From the same arguments, we have the following expression of $\ell_n(\hat{\theta}_1, \hat{\theta}_2)$:

$$\begin{aligned}
\ell_n(\hat{\theta}_1, \hat{\theta}_2) &= \ell_n(\theta_0^*, \theta_0^*) + \frac{1}{2}\nabla \ell_n(\theta_0^*, \theta_0^*)^\top \left(-\nabla^2 \ell_n(\theta_0^*, \theta_0^*)\right)^{-1} \nabla \ell_n(\theta_0^*, \theta_0^*) + o_{\mathbb{P}}(1) \\
&= \ell_n(\theta_0^*, \theta_0^*) \\
&\quad + \frac{1}{2}\left[\left(-\frac{1}{n_1}\sum_{i=1}^{n_1} \nabla^2 \log\left(\mathbb{P}(X_i; \theta_0^*)\right)\right)^{-1/2} \frac{1}{\sqrt{n_1}} \sum_{i=1}^{n_1} \nabla \log\left(\mathbb{P}(X_i; \theta_0^*)\right)\right]^{\otimes 2} \\
&\quad + \frac{1}{2}\left[\left(-\frac{1}{n-n_1}\sum_{i=n_1+1}^{n} \nabla^2 \log\left(\mathbb{P}(X_i; \theta_0^*)\right)\right)^{-1/2} \frac{1}{\sqrt{n-n_1}} \sum_{i=n_1+1}^{n} \nabla \log\left(\mathbb{P}(X_i; \theta_0^*)\right)\right]^{\otimes 2} \\
&\quad + o_{\mathbb{P}}(1).
\end{aligned}$$

From the consistency of

$$\frac{1}{n_1}\sum_{i=1}^{n_1} \nabla^2 \log\left(\mathbb{P}(X_i; \theta_0^*)\right), \quad \frac{1}{n-n_1}\sum_{i=n_1+1}^{n} \nabla^2 \log\left(\mathbb{P}(X_i; \theta_0^*)\right), \quad \frac{1}{n}\sum_{i=1}^{n} \nabla^2 \log\left(\mathbb{P}(X_i; \theta_0^*)\right)$$

towards $I(\theta_0^*)$ we can replace $\sum_{i=1}^{n_1} \nabla^2 \log\left(\mathbb{P}(X_i; \theta_0^*)\right)/n_1$ and $\sum_{i=n_1+1}^{n} \nabla^2 \log\left(\mathbb{P}(X_i; \theta_0^*)\right)/(n-n_1)$ by $\hat{I}(\theta_0^*)$ in the above equation. Taking the difference between $2\ell_n(\hat{\theta}_1, \hat{\theta}_2)$ and $2\tilde{\ell}_n(\hat{\theta}_0)$ we conclude using the consistency of $\hat{\theta}$ towards θ_0.

A.4.2 Proof of Theorem 2

The proofs of 1. and 2. of the theorem are identical, therefore only the proof of 1. is presented. We first write:

$$\begin{aligned}
\ell_n(\hat{\theta}_1, \hat{\theta}_2) &= \sum_{i=1}^{n_1} \log \mathbb{P}(X_i; \hat{\theta}_1) + \sum_{i=n_1+1}^{n} \log \mathbb{P}(X_i; \hat{\theta}_2) \\
&= \sum_{i=1}^{n_1} \log \mathbb{P}(X_i; \hat{\theta}_1) + \sum_{i=n_1+1}^{n} \log \mathbb{P}(X_i; \theta_0^*) \\
&\quad + \frac{1}{2}\left[\left(-\frac{1}{n}\sum_{i=1}^{n} \nabla^2 \log\left(\mathbb{P}(X_i; \theta_0^*)\right)\right)^{-1/2} \frac{1}{\sqrt{n-n_1}} \sum_{i=n_1+1}^{n} \nabla \log\left(\mathbb{P}(X_i; \theta_0^*)\right)\right]^{\otimes 2} \\
&\quad + o_{\mathbb{P}}(1),
\end{aligned}$$

where we used a similar argument as in Equation (A.9) and we replaced $\sum_{i=n_1+1}^{n} \nabla^2 \log\left(\mathbb{P}(X_i; \theta_0^*)\right)/(n-n_1)$ by $\hat{I}(\theta_0^*)$. Since n_1 is fixed, the sum $\sum_{i=n_1+1}^{n} \nabla \log\left(\mathbb{P}(X_i; \theta_0^*)\right)$ can be replaced by the sum $\sum_{i=1}^{n} \nabla \log\left(\mathbb{P}(X_i; \theta_0^*)\right)$ using the fact that $\sum_{i=1}^{n_1} \nabla \log\left(\mathbb{P}(X_i; \theta_0^*)\right)/\sqrt{n-n_1}$ tends towards 0 in probability. We finally get:

$$\begin{aligned}
\ell_n(\hat{\theta}_1, \hat{\theta}_2) &= \sum_{i=1}^{n_1} \log \mathbb{P}(X_i; \hat{\theta}_1) + \sum_{i=n_1+1}^{n} \log \mathbb{P}(X_i; \theta_0^*) \\
&\quad + \frac{1}{2}\left[\left(-\frac{1}{n}\sum_{i=1}^{n} \nabla^2 \log\left(\mathbb{P}(X_i; \theta_0^*)\right)\right)^{-1/2} \frac{1}{\sqrt{n}} \sum_{i=1}^{n} \nabla \log\left(\mathbb{P}(X_i; \theta_0^*)\right)\right]^{\otimes 2} + o_{\mathbb{P}}(1).
\end{aligned} \tag{A.10}$$

Next, we write

$$\tilde{\ell}_n(\hat{\theta}_0) = \sum_{i=1}^{n_1} \log \mathbb{P}(X_i; \hat{\theta}_0) + \sum_{i=n_1+1}^{n} \log \mathbb{P}(X_i; \hat{\theta}_0)$$

and

$$\sum_{i=n_1+1}^{n} \log \mathbb{P}(X_i;\theta_0^*) = \sum_{i=n_1+1}^{n} \log \mathbb{P}(X_i;\hat{\theta}_0) + (\hat{\theta}_0 - \theta^*)^\top \sum_{i=n_1+1}^{n} \nabla \log \mathbb{P}(X_i;\hat{\theta}_0) + \frac{1}{2}(\theta_0^* - \hat{\theta}_0)^\top \sum_{i=n_1+1}^{n} \nabla^2 \log \mathbb{P}(X_i;\theta_0'),$$

where θ_0' is on the real line between θ_0^* and $\hat{\theta}_0$. Since n_1 is fixed and $\sum_{i=1}^{n} \nabla \log \mathbb{P}(X_i;\hat{\theta}_0) = 0$, we have

$$(\hat{\theta}_0 - \theta^*)^\top \sum_{i=n_1+1}^{n} \nabla \log \mathbb{P}(X_i;\hat{\theta}_0) = -(\hat{\theta}_0 - \theta^*)^\top \sum_{i=1}^{n_1} \nabla \log \mathbb{P}(X_i;\hat{\theta}_0) = o_{\mathbb{P}}(1).$$

From Equation (A.8) and using the same arguments as in the development of Equation (A.9), we finally have:

$$\begin{aligned}
\tilde{\ell}_n(\hat{\theta}_0) &= \sum_{i=1}^{n_1} \log \mathbb{P}(X_i;\hat{\theta}_0) + \sum_{i=n_1+1}^{n} \log \mathbb{P}(X_i;\theta_0^*) - \frac{1}{2}(\theta_0^* - \hat{\theta}_0)^\top \sum_{i=n_1+1}^{n} \nabla^2 \log \mathbb{P}(X_i;\theta_0') \\
&= \sum_{i=1}^{n_1} \log \mathbb{P}(X_i;\hat{\theta}_0) + \sum_{i=n_1+1}^{n} \log \mathbb{P}(X_i;\theta_0^*) \\
&\quad + \frac{1}{2}\frac{n-n_1}{n} \left[\left(-\frac{1}{n} \sum_{i=1}^{n} \nabla^2 \log \left(\mathbb{P}(X_i;\theta_0^*) \right) \right)^{-1/2} \frac{1}{\sqrt{n}} \sum_{i=1}^{n} \nabla \log \left(\mathbb{P}(X_i;\theta_0^*) \right) \right]^{\otimes 2} + o_{\mathbb{P}}(1),
\end{aligned}$$

where we replaced $\sum_{i=n_1}^{n} \nabla^2 \log \left(\mathbb{P}(X_i;\theta_0^*) \right) / (n - n_1)$ by $\sum_{i=1}^{n} \nabla^2 \log \left(\mathbb{P}(X_i;\theta_0^*) \right) / n$ in the above expression. Taking the difference between Equation (A.10) and the last equation gives the desired result.

References

[1] Robert J Shiller. *Market volatility*. MIT press, 1992.

[2] A Ronald Gallant, David Hsieh, and George Tauchen. Estimation of stochastic volatility models with diagnostics. *Journal of econometrics*, 81(1):159–192, 1997.

[3] Haeran Cho and Piotr Fryzlewicz. Multiscale and multilevel technique for consistent segmentation of nonstationary time series. *Statistica Sinica*, pages 207–229, 2012.

[4] Rebecca Killick, Paul Fearnhead, and Idris A Eckley. Optimal detection of changepoints with a linear computational cost. *Journal of the American Statistical Association*, 107(500):1590–1598, 2012.

[5] Jaxk Reeves, Jien Chen, Xiaolan L Wang, Robert Lund, and Qi Qi Lu. A review and comparison of changepoint detection techniques for climate data. *Journal of applied meteorology and climatology*, 46(6):900–915, 2007.

[6] Rebecca Killick, Idris A Eckley, Kevin Ewans, and Philip Jonathan. Detection of changes in variance of oceanographic time-series using changepoint analysis. *Ocean Engineering*, 37(13):1120–1126, 2010.

[7] Marie Gomot, Frédéric A Bernard, Matthew H Davis, Matthew K Belmonte, Chris Ashwin, Edward T Bullmore, and Simon Baron-Cohen. Change detection in children with autism: an auditory event-related fmri study. *Neuroimage*, 29(2):475–484, 2006.

[8] Matthew H Davis and Ingrid S Johnsrude. Hearing speech sounds: top-down influences on the interface between audition and speech perception. *Hearing research*, 229(1-2):132–147, 2007.

[9] ES Venkatraman and Adam B Olshen. A faster circular binary segmentation algorithm for the analysis of array cgh data. *Bioinformatics*, 23(6):657–663, 2007.

[10] Ronglai Shen, Adam B Olshen, and Marc Ladanyi. Integrative clustering of multiple genomic data types using a joint latent variable model with application to breast and lung cancer subtype analysis. *Bioinformatics*, 25(22):2906–2912, 2009.

[11] ES Venkatraman and Adam B Olshen. Dnacopy: a package for analyzing dna copy data. *Department of Epidemiology and Biostatistics. Memorial Sloan-Kettering Cancer Center*, 2007.

[12] Nancy R Zhang, David O Siegmund, Hanlee Ji, and Jun Z Li. Detecting simultaneous changepoints in multiple sequences. *Biometrika*, 97(3):631–645, 2010.

[13] Brad Jackson, Jeffrey D Scargle, David Barnes, Sundararajan Arabhi, Alina Alt, Peter Gioumousis, Elyus Gwin, Paungkaew Sangtrakulcharoen, Linda Tan, and Tun Tao Tsai. An algorithm for optimal partitioning of data on an interval. *IEEE Signal Processing Letters*, 12(2):105–108, 2005.

[14] Robert Maidstone, Toby Hocking, Guillem Rigaill, and Paul Fearnhead. On optimal multiple changepoint algorithms for large data. *Statistics and computing*, 27:519–533, 2017.

[15] G Rigaill, T Hocking, R Maidstone, and P Fearnhead. fpop: Segmentation using optimal partitioning and function pruning. *R package*, 2019.

[16] Toby Dylan Hocking, Guillem Rigaill, Paul Fearnhead, and Guillaume Bourque. Constrained dynamic programming and supervised penalty learning algorithms for peak detection in genomic data. *Journal of Machine Learning Research*, 21(87):1–40, 2020.

[17] Vincent Runge, Toby Dylan Hocking, Gaetano Romano, Fatemeh Afghah, Paul Fearnhead, and Guillem Rigaill. gfpop: an r package for univariate graph-constrained change-point detection. *Journal of Statistical Software*, 106(6), 2023.

[18] Olivier Bouaziz and Grégory Nuel. A change-point model for detecting heterogeneity in ordered survival responses. *Statistical methods in medical research*, 27(12):3595–3611, 2018.

[19] Flora Alarcon and Gregory Nuel. Detecting latent exposure in genome-wide association studies using a breakpoint model for logistic regression. *Statistical methods in medical research*, 28(6):1781–1792, 2019.

[20] Gilles Celeux and Gérard Govaert. A classification em algorithm for clustering and two stochastic versions. *Computational statistics & Data analysis*, 14(3):315–332, 1992.

[21] Robert Tibshirani, Michael Saunders, Saharon Rosset, Ji Zhu, and Keith Knight. Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 67(1):91–108, 2005.

[22] Alessandro Rinaldo. Properties and refinements of the fused lasso. 2009.

[23] Andrew Jhon Scott and Martin Knott. A cluster analysis method for grouping means in the analysis of variance. *Biometrics*, pages 507–512, 1974.

[24] Olivier Bouaziz and Agathe Guilloux. A penalized algorithm for event-specific rate models for recurrent events. *Biostatistics*, 16(2):281–294, 2015.

[25] John D Kalbfleisch and Ross L Prentice. *The statistical analysis of failure time data*. John Wiley & Sons, 2011.

[26] Trevor Hastie and Werner Stuetzle. Principal curves. *Journal of the American statistical association*, 84(406):502–516, 1989.