



**HAL**  
open science

## Segmentation in the mean of heteroscedastic data via cross-validation

Sylvain Arlot, Alain Celisse

► **To cite this version:**

Sylvain Arlot, Alain Celisse. Segmentation in the mean of heteroscedastic data via cross-validation. 2009. hal-00363627v1

**HAL Id: hal-00363627**

**<https://hal.science/hal-00363627v1>**

Submitted on 23 Feb 2009 (v1), last revised 8 Apr 2009 (v2)

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Segmentation in the mean of heteroscedastic data via cross-validation

Sylvain Arlot and Alain Celisse

February 23, 2009

## Abstract

This paper tackles the problem of detecting abrupt changes in the mean of a heteroscedastic signal by model selection, without knowledge on the variations of the noise. Whereas most existing methods are not robust to heteroscedasticity, a new family of algorithms is proposed showing that cross-validation methods can be successful in this framework. The robustness to heteroscedasticity of the new cross-validation based change-point detection algorithms is supported by an extensive simulation study, together with recent theoretical results. An application to comparative genomic hybridization data is provided, showing that robustness to heteroscedasticity can indeed be required for their analysis.

## 1 Introduction

### 1.1 Change-point detection

The change-point detection problem, also called one-dimensional segmentation, deals with a stochastic process the distribution of which abruptly changes at some unknown instants. The purpose is to recover the location of these changes. This problem is motivated by a wide range of applications, from voice recognition and time-series analysis in the financial area [34], to biology and CGH (Comparative Genomic Hybridization) data analysis [40]. A huge amount of paper exist about change-point detection, in many frameworks; we refer to the books by Basseville and Nikiforov [14] and by Brodsky and Darkhovsky [19] for a complete bibliography.

The first papers on change-point detection were devoted to the search for the location of a unique change-point (also named breakpoint; see [39] for instance). Looking for multiple change-points is a harder task and has

been studied later. For instance, Yao [54] used the BIC criterion for detecting multiple change-points in a Gaussian signal, and Miao and Zhao [38] proposed an approach relying on rank statistics.

The setting of the paper is the following. The values  $Y_1, \dots, Y_n \in \mathbb{R}$  of a noisy signal at points  $t_1, \dots, t_n$  are observed, with

$$Y_i = s(t_i) + \sigma(t_i)\epsilon_i, \quad \mathbb{E}[\epsilon_i] = 0 \quad \text{and} \quad \text{Var}(\epsilon_i) = 1. \quad (1)$$

The function  $s$  is called the *regression function* and is assumed to be piecewise-constant, or at least well approximated by piecewise constant functions, that is  $s$  is smooth everywhere except at a few breakpoints. The noise terms  $\epsilon_1, \dots, \epsilon_n$  are assumed to be independent and identically distributed. No assumption is made on  $\sigma : [0, 1] \mapsto [0, \infty)$ .

The present work belongs to the *off-line* setting, where data  $(t_i, Y_i)_{1 \leq i \leq n}$  are entirely observed before detecting the change-points. This setting differs from the *on-line* setting, where change-points have to be detected in a signal observed sequentially.

As pointed out by Lavielle [33], multiple change-point detection algorithms generally tackle one among the following three problems:

1. Detecting changes in the mean  $s(t_i)$  assuming the standard-deviation  $\sigma$  is constant,
2. Detecting changes in the standard-deviation  $\sigma$  assuming the mean  $s(t_i)$  is constant,
3. Detecting changes in the whole distribution, with no distinction between changes in the mean  $s(t_i)$  and changes in the standard-deviation  $\sigma$ .

In applications such as CGH data analysis, changes in the mean  $s(t_i)$  have an important biological meaning, since they correspond to the limits of amplified or deleted areas of chromosomes. However, the assumption made in problem 1 above that the standard-deviation  $\sigma$  is constant does not always hold. See Section 6 for more details on CGH data, where heterogeneities in variance correspond to experimental artefacts that we would like to remove.

Therefore, CGH data analysis requires to solve a fourth problem, which is the purpose of the present article:

4. Detecting changes in the mean  $s(t_i)$  with a general standard-deviation  $\sigma : [0, 1] \mapsto [0, \infty)$ .

Compared to problem 1, the difference is the presence of an additional nuisance parameter  $\sigma$  making problem 4 more difficult. Up to the best of our knowledge, no change-point detection algorithm has ever been proposed for solving problem 4 with *no information on  $\sigma$* .

## 1.2 Model selection

Model selection is a successful approach for multiple change-point detection, as shown by the works of Lavielle [33] and of Lebarbier [35] for instance. Indeed, a set of change-points—called a segmentation—is naturally associated with the set of piecewise-constant functions that may only jump at these change-points. Given a set of functions (called a model), estimation can be performed by minimizing the least-squares criterion. Therefore, detecting changes in the mean of a signal amounts to select such a model.

More precisely, given a collection of models  $\{S_m\}_{m \in \mathcal{M}_n}$  and the associated collection of least-squares estimators  $\{\hat{s}_m\}_{m \in \mathcal{M}_n}$ , the purpose of model selection is to provide a model index  $\hat{m}$  such that  $\hat{s}_{\hat{m}}$  reaches the “best performance” among all the estimators  $\{\hat{s}_m\}_{m \in \mathcal{M}_n}$ .

Model selection can target two different goals. On the one hand, a procedure is *efficient* when its quadratic risk is smaller than the smallest quadratic risk of the estimators  $\{\hat{s}_m\}_{m \in \mathcal{M}_n}$ , up to a constant factor  $C_n \geq 1$ . Such a property is called an *oracle inequality* when it holds for every finite sample size; this property is called *asymptotic efficiency* when it holds with a constant 1, but only when  $n$  tends to infinity. Asymptotic efficiency is the goal of AIC [2, 3] and Mallows’  $C_p$  [37], among many others.

On the other hand, assuming that  $s$  belongs to one of the models  $\{S_m\}_{m \in \mathcal{M}_n}$ , a procedure is *model consistent* when it chooses the smallest model containing  $s$  asymptotically with probability one. Model consistency is the goal of BIC [45] for instance. See also the article by [52] about the distinction between efficiency and model consistency.

In the present paper as in the one by Lebarbier [35], the quality of a multiple change-point detection procedure is assessed by the quadratic risk, so that *a change in the mean hidden by the noise should not be detected*. This choice is motivated by applications where the signal-to-noise ratio is not large, for which recovering every true change-point is hopeless. Therefore, *efficient* model selection procedures will be used in order to detect the change-points.

Without *a priori* information on the locations of the change-points, the natural collection of models for change-point detection depends on the sam-

ple size  $n$ . Indeed, with  $n$  points in the design, there are  $\binom{n}{D}$  different partitions of  $[0, 1]$  into  $D$  intervals, each partition corresponding to a set of  $D-1$  change-points. Since  $D$  can take any value between 1 and  $n$ ,  $\text{Card}(\mathcal{M}_n)$  is of order  $2^n$ . Therefore, the model selection procedures used for multiple change-point detection have to satisfy *non-asymptotic* oracle inequalities: the collection of models cannot be assumed to be fixed with the sample size  $n$  tending to infinity.

Most model selection results consider “polynomial” collections of models  $\mathcal{M}_n$ , that is when  $\text{Card}(\mathcal{M}_n) \leq Cn^\alpha$  for some constants  $C, \alpha \geq 0$ . For polynomial collections, procedures like AIC or Mallows’  $C_p$  are proved to satisfy oracle inequalities in various frameworks [10, 17, 11, 18], assuming that data are *homoscedastic*, that is  $\sigma$  remains constant for every  $t_i$ s.

However as shown in [7], the same procedures are suboptimal when data are *heteroscedastic*, that is the variance is non-constant, because they are penalizing models proportionally to their number of parameters. Therefore, other penalties must be used. For instance, resampling penalization is optimal with heteroscedastic data [6]. Another approach has been explored by Gendre [27], which consists in estimating simultaneously the mean and the variance, but his definition is restricted to a particular collection of models.

When the cardinality of  $\mathcal{M}_n$  is larger than polynomial (as for change-point detection, where it is “exponential”, that is of order  $\exp(\alpha n)$  for some  $\alpha > 0$ ), the above penalization procedures fail. Indeed, Birgé and Massart [18] proved that the minimal amount of penalization required for a procedure to satisfy an oracle inequality is of the form

$$\text{pen}(m) = c_1 \frac{D_m}{n} + c_2 \frac{D_m}{n} \log \left( \frac{n}{D_m} \right) , \quad (2)$$

where  $c_1$  and  $c_2$  are positive constants. Optimal values for  $c_1$  and  $c_2$  have been obtained by simulations by Lebarbier [35] in the context of change-point detection. Similar penalties have been proposed independently by other authors [44, 1, 12, 51] and are shown to provide satisfactory results.

Nevertheless, all these results assume that data are homoscedastic. Actually, the model selection problem with heteroscedastic data and an exponential collection has never been considered in the model selection literature, up to the best of our knowledge.

Having in mind that any penalty proportional to  $D_m$  fails for heteroscedastic data and polynomial collections of models [7], the penalty (2) can be conjectured to be suboptimal for model selection over an exponential collec-

tion of models, since (2) is very close to be proportional to  $D_m$  (at least for small values of  $D_m$ ).

Following results proved in [6, 8], resampling methods turn out to be quite natural with heteroscedastic data, in particular cross-validation. The purpose of the present work is to show how resampling strategies such as cross-validation can be fruitfully used for detecting changes in the mean of heteroscedastic data.

### 1.3 Cross-validation and resampling

Resampling refers to a general principle consisting in generating new samples (*re-samples*) from the original data, in order to infer some features of the underlying distribution of the observations. Well known examples of resampling algorithms are the jackknife [43], the bootstrap [24], as well as cross-validation algorithms such as the leave-one-out [31, 49] (Loo) and  $V$ -fold cross-validation (VFCV) [25, 26].

On the positive side, resampling strategies are known to adapt to a wide range of statistical settings, from density estimation [48, 22] to regression [50, 53] and learning [30, 13]. On the negative side, most theoretical results on resampling algorithms are asymptotic nature [36, 28, 29, 46], whereas model selection among a collection of models  $\{S_m\}_{m \in \mathcal{M}_n}$  depending on the sample size  $n$  requires to use the non-asymptotic point of view, as in [5, 21] for instance.

When data are heteroscedastic, resampling penalization enjoys some optimality properties [7] for selecting among some polynomial collections of models, as noticed in Section 1.2. However, to the best of our knowledge, nearly nothing is known on resampling algorithms when the collection of models is rich and data are heteroscedastic, as in our setting. From the literature on model selection and resampling [16, 18, 23], we can only conclude that the Loo estimator of the quadratic risk over all possible partitions is not appropriate since it leads to overfitting.

### 1.4 Contributions of the paper

The main purpose of the present work is to design a resampling-based model selection algorithm (Algorithm 6) that can be used for detecting multiple changes in the mean of a heteroscedastic signal. Therefore, such an algorithm has to adapt to heteroscedasticity when the collection of models is exponential, which has never been obtained before. In particular, the goal is to obtain an alternative to Birgé and Massart's penalization procedure

[17] which is reliable with heteroscedastic data.

Another major difficulty tackled in this paper is the computational cost of resampling methods when selecting among  $2^n$  models. Even when the number  $D - 1$  of change-points is given, exploring the  $\binom{n-1}{D-1}$  partitions of  $[0, 1]$  into  $D$  intervals and performing a resampling algorithm for each partition is not feasible when  $n$  is large and  $D > 0$ . The contribution of the paper relies on the combination of closed-form formulas for Leave- $p$ -out (Lpo) estimates of the risk obtained in [23], dynamic programming, and  $V$ -fold cross-validation.

Several results in this work provide a deeper understanding of the least-squares minimization algorithm applied to heteroscedastic data. Even if this algorithm is actually widely used in model selection literature with constant variance, nearly nothing is known about it under heteroscedasticity. Another contribution of this work lies in some recent theoretical results describing the behaviour of least-squares minimization compared with that of Leave- $p$ -out.

The paper is organized as follows. The statistical framework is described in Section 2. First, the problem of selecting the “best” segmentation given the number of change-points is tackled in Section 3. Theoretical results and an extensive simulation study shows that the usual minimization of the least-squares criterion can be misleading when data are heteroscedastic, whereas cross-validation algorithms provide satisfactory results in the same framework.

Then, the problem of choosing the number of breakpoints from the data is addressed in Section 4. As supported by an extensive simulation study,  $V$ -fold cross-validation (VFCV) leads to a computationally feasible and statistically efficient method when data are heteroscedastic, contrary to algorithms implicitly assuming homoscedasticity.

The resampling methods of Sections 3 and 4 are combined in Section 5, leading to a family of resampling-based algorithms for detecting changes in the mean of an heteroscedastic signal. A wide simulation study shows they perform well with both homoscedastic and heteroscedastic data, improving significantly the performance of algorithms implicitly assuming homoscedasticity.

Finally, Section 6 illustrates on a real data set the promising behaviour of the proposed algorithms for analyzing CGH microarray data, compared to the algorithms previously used in this setting.

## 2 Statistical framework

### 2.1 Regression on a fixed design

Let  $t_1 < \dots < t_n \in [0, 1]$  be some deterministic observation points,  $s : [0, 1] \mapsto \mathbb{R}$  and  $\sigma : [0, 1] \mapsto [0, \infty)$  be some functions and define

$$\forall i \in \{1, \dots, n\}, \quad Y_i = s(t_i) + \sigma(t_i)\epsilon_i, \quad (3)$$

where  $\epsilon_1, \dots, \epsilon_n$  are independent and identically distributed random variables with  $\mathbb{E}[\epsilon_i] = 0$  and  $\mathbb{E}[\epsilon_i^2] = 1$ .

As explained in Section 1.1, the goal is to find a (piecewise-constant) function  $f : [0, 1] \mapsto \mathbb{R}$  close to  $s$  in terms of the quadratic loss

$$\|s - f\|_n^2 := \frac{1}{n} \sum_{i=1}^n (f(t_i) - s(t_i))^2.$$

### 2.2 Least-squares estimator

A classical estimator of  $s$  is the *least-squares estimator*, defined as follows. For every  $f : [0, 1] \mapsto \mathbb{R}$ , the least-squares criterion at  $f$ , is defined by

$$P_n \gamma(f) := \frac{1}{n} \sum_{i=1}^n (Y_i - f(t_i))^2.$$

The notation  $P_n \gamma(f)$  means that the function  $(t, Y) \mapsto \gamma(f; (t, Y)) := (Y - f(t))^2$  is integrated with respect to the empirical distribution  $P_n := n^{-1} \sum_{i=1}^n \delta_{(t_i, Y_i)}$ .  $P_n \gamma(f)$  is also called the *empirical risk* of  $f$ .

Then, given a set  $S$  of functions  $[0, 1] \mapsto \mathbb{R}$  (called a *model*), the minimizer of the empirical risk over  $S$  is the least-squares estimator  $\text{ERM}(S, P_n)$ , that is

$$\text{ERM}(S, P_n) := \text{Argmin}_{f \in S} \{P_n \gamma(f)\}.$$

The notation  $\text{ERM}(S, P_n)$  stresses that the empirical risk minimization algorithm takes a model  $S$  and a data sample as inputs.

When a collection of models  $\{S_m\}_{m \in \mathcal{M}_n}$  is given,  $\hat{s}_m$  and  $\hat{s}_m(P_n)$  are shortcuts for  $\text{ERM}(S_m, P_n)$ .

### 2.3 Collection of models

Since the goal is to detect jumps of  $s$ , the models considered in this article are the sets of piecewise constant functions with respect to some partitions of  $[0, 1]$ .



For every sequence of integers  $0 < \alpha_1 < \alpha_2 < \dots < \alpha_K < n$  (the breakpoints),  $(I_\lambda)_{\lambda \in \Lambda(\alpha_1, \dots, \alpha_K)}$  denotes the partition

$$[0; t_{\alpha_1}), [t_{\alpha_1}; t_{\alpha_2}), \dots, [t_{\alpha_K}; 1] .$$

Then, the model  $S_{(\alpha_1, \dots, \alpha_K)}$  is defined as the set of piecewise constant functions that can only jump at one of the  $t_{\alpha_j}$ s.

For every  $K \in \{0, \dots, n-1\}$ , let  $\mathcal{M}_n(K+1)$  denote the set of such sequences  $(\alpha_1, \dots, \alpha_K)$  of length  $K$ , so that  $\{S_m\}_{m \in \mathcal{M}_n(K+1)}$  is the collection of models of piecewise constant functions with  $K$  breakpoints. When  $K = 0$ ,  $\mathcal{M}_n(1) = \{\emptyset\}$  and the model  $S_\emptyset$  is the linear space of constant functions on  $[0, 1]$ . Remark that for every  $K$  and  $m \in \mathcal{M}_n(K+1)$ ,  $S_m$  is a vector space of dimension  $D_m = K+1$ .

The collection of models considered in this article can now be defined as  $\{S_m\}_{m \in \mathcal{M}_n}$ , where  $\mathcal{M}_n = \bigcup_{1 \leq D \leq n} \mathcal{M}_n(D)$ . This collection has a cardinality  $2^{n-1}$ .

REMARK: Since a model  $S_m$  is uniquely defined by its index  $m$ , we also call  $m$  a model.

## 2.4 Model selection

Among all the models  $\{S_m\}_{m \in \mathcal{M}_n}$ , the best one is defined as the minimizer of the *quadratic loss*  $\|s - \hat{s}_m\|_n^2$  over  $m \in \mathcal{M}_n$  and called the *oracle*  $m^*$ . Since the oracle depends on  $s$ , one can only expect to select  $\hat{m}(P_n)$  from the data such that the quadratic loss of  $\hat{s}_{\hat{m}}$  is close to that of the oracle with high probability, that is

$$\|s - \hat{s}_{\hat{m}}\|_n^2 \leq C \inf_{m \in \mathcal{M}_n} \left\{ \|s - \hat{s}_m\|_n^2 \right\} + R_n \quad (4)$$

where  $C$  is close to 1 and  $R_n$  is a small remainder term (typically of order  $n^{-1}$ ).

Inequality (4) is called an *oracle inequality*. A procedure satisfying (4) with  $C = C_n$  tending to one when  $n$  tends to infinity and  $R_n$  negligible is called *efficient*, or asymptotically efficient.

## 3 Breakpoint locations

A usual strategy for multiple change-point detection [33, 35] is to dissociate the search for the best segmentation given the number of breakpoints from the choice of this number of breakpoints.

In this section, the number  $K$  of breakpoints is considered as known and the goal is to find the location of the change-points.

### 3.1 Empirical risk minimization’s failure with heteroscedastic data

As explained by Lavielle [32] among many authors, minimizing the least-squares criterion over  $\{\widehat{s}_m\}_{m \in \mathcal{M}(K+1)}$  is a reliable way of getting the best segmentation with  $K$  change-points. This leads to the following algorithm:

**Algorithm 1.**

$$\widehat{m}_{\text{ERM}}(D) = \text{Argmin}_{m \in \mathcal{M}_n(D)} P_n \gamma(\widehat{s}_m) = \text{ERM}(\widetilde{S}_D, P_n),$$

where  $D = K + 1$  and

$$\widetilde{S}_{K+1} := \cup_{m \in \mathcal{M}(K+1)} S_m.$$

is the set of piecewise constant functions with exactly  $K$  change-points, chosen among  $t_1, \dots, t_n$ .

REMARK: Dynamic programming [15] leads to an efficient implementation of algorithm 1 with computational complexity of order  $\mathcal{O}(n^2)$ .

Since the number of change-points  $K$  is fixed, the “oracle” model can now be defined as

$$m^*(D) = \text{Argmin}_{m \in \mathcal{M}_n(D)} \left\{ \|s - \widehat{s}_m\|_n^2 \right\} .$$

Figure 1 illustrates how far  $\widehat{m}_{\text{ERM}}(D)$  is from  $m^*(D)$  according to variations of the standard-deviation  $\sigma$ . On the one hand, when data are homoscedastic, empirical risk minimization yields a segmentation close to the oracle (Figure 1, left). On the other hand, when data are heteroscedastic, empirical risk minimization introduces artificial breakpoints in regions where the noise-level is high and misses breakpoints in regions where the noise-level is lower (Figure 1, right).

The failure of empirical risk minimization with heteroscedastic data observed on Figure 1 can be explained by Lemma 1 below. Indeed, the leading term of the criterions  $P_n \gamma(\widehat{s}_m)$  and  $\|s - \widehat{s}_m\|_n^2$ , respectively minimized by  $\widehat{m}_{\text{ERM}}(D)$  and  $m^*(D)$  over  $\mathcal{M}_n(D)$ , are their expectations, as proved by the concentration inequalities of [8, Proposition 9] for instance.

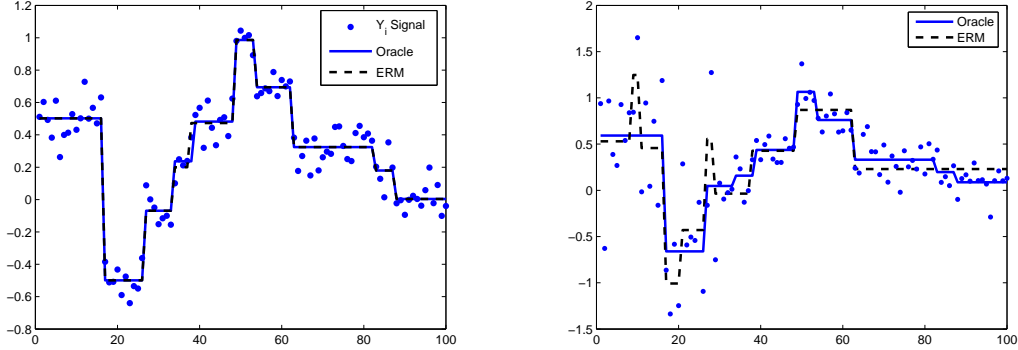


Figure 1: Comparison of  $\widehat{s}_{\widehat{m}_{\text{ERM}}(D)}$  (dashed line) and  $\widehat{s}_{m^*(D)}$  (dotted line) with  $D = 10$ . Data are generated as described in Section 3.3.1 with  $n = 100$  data points;  $s = s_3$  is piecewise constants with 9 breakpoints;  $\sigma$  is either constant (left,  $\sigma = \sigma_c$ ) or piecewise constant (right,  $\sigma = \sigma_{pc,3}$ ).

**Lemma 1.** Let  $m \in \mathcal{M}_n$  and define  $s_m = \text{Argmin}_{f \in S_m} \|s - f\|_n^2$ . Then, for every  $m \in \mathcal{M}_n$ ,

$$\mathbb{E}[P_n \gamma(\widehat{s}_m)] = \|s - s_m\|_n^2 - \frac{1}{n} \sum_{\lambda \in \Lambda_m} (\sigma_\lambda^r)^2 + \frac{1}{n} \sum_{i=1}^n \sigma(t_i)^2 \quad (5)$$

$$\mathbb{E}[\|s - \widehat{s}_m\|_n^2] = \|s - s_m\|_n^2 + \frac{1}{n} \sum_{\lambda \in \Lambda_m} (\sigma_\lambda^r)^2 \quad (6)$$

where for every  $\lambda \in \Lambda_m$ ,

$$(\sigma_\lambda^r)^2 := \frac{1}{\text{Card}(\{k \mid t_k \in I_\lambda\})} \sum_{i=1}^n \sigma(t_i)^2 \mathbf{1}_{t_i \in I_\lambda} .$$

Lemma 1 is proved in [23].

As it is well-known in the model selection literature, the expectation of the quadratic loss (6) is the sum of two terms:  $\|s - s_m\|_n^2$  is the bias of the model  $S_m$ , and

$$V(m) =: n^{-1} \sum_{\lambda \in \Lambda_m} (\sigma_\lambda^r)^2$$

is a variance term, measuring the difficulty of estimating the  $D_m$  parameters of the model  $S_m$ . Up to the term  $n^{-1} \sum_{i=1}^n \sigma(t_i)^2$  which does not depend on  $m$ , the empirical risk underestimates the quadratic risk (*i.e.* the expectation of the quadratic loss), as shown by (5), because of the variance term  $V(m)$ .

Nevertheless, when data are homoscedastic, that is when  $\forall i, \sigma(t_i) = \sigma$ ,

$$V(m) = \frac{1}{n} \sum_{\lambda \in \Lambda_m} (\sigma_\lambda^r)^2 = \frac{D_m \sigma^2}{n}$$

is the same for all  $m \in \mathcal{M}_n(D)$ . Therefore, (5) and (6) show that for every  $D \geq 1$ ,

$$\text{Argmin}_{m \in \mathcal{M}_n(D)} \{\mathbb{E}[P_n \gamma(\widehat{s}_m)]\} = \text{Argmin}_{m \in \mathcal{M}_n(D)} \left\{ \mathbb{E} \left[ \|s - \widehat{s}_m\|_n^2 \right] \right\} .$$

Hence,  $\widehat{m}_{\text{ERM}}(D)$  and  $m^*(D)$  tend to be close to one another, as on the left of Figure 1.

On the contrary, when data are heteroscedastic, the variance term  $V(m)$  can be quite different among the models  $\{S_m\}_{m \in \mathcal{M}_n(D)}$ , even though they have the same number of parameters. Indeed,  $V(m)$  increases when a breakpoint is moved from an area where  $\sigma$  is small to an area where  $\sigma$  is large. Therefore, the empirical risk minimization algorithm rather puts breakpoints in noisy regions in order to minimize  $-V(m)$  in (5). This is illustrated in the right panel of Figure Figure 1, where the oracle segmentation  $m^*(D)$  has more breakpoints in regions where  $\sigma$  is small.

## 3.2 Cross-validation

Cross-validation (CV) methods are natural candidates for fixing the failure of empirical risk minimization when data are heteroscedastic. Indeed, CV and related methods are naturally adaptive to heteroscedasticity [8]. Therefore, the segmentation with  $D - 1$  breakpoints chosen by minimizing of the CV estimator of the quadratic risk should be close to the oracle  $m^*(D)$ .

### 3.2.1 Heuristics

The cross-validation heuristics [4, 49] relies on a data splitting idea: For each candidate algorithm (for instance, among  $\{\text{ERM}(S_m; \cdot)\}_{m \in \mathcal{M}_n(D)}$ ), a part of the data—called training set—is used for training the algorithm, and the rest of the data—called validation set—is used to estimate the risk of the segmentation output by the algorithm. This simple strategy is called validation or hold-out. One can also split data several times and average the estimated values of the risk over the splits; such a strategy is called *cross-validation* (CV).

Usually, CV is used for estimating the risk of an estimator trying to “predict” a new observation  $(T, Y)$ , that is a random variable independent

from  $(t_i, Y_i)_{1 \leq i \leq n}$  and distributed as each of the  $(t_i, Y_i)$ . For instance, in the least-squares regression setting, the prediction risk of  $\hat{s} = \hat{s}(P_n)$  would be

$$\mathbb{E}_{(T,Y)} \left[ (\hat{s}(T) - Y)^2 \right] .$$

Nevertheless, in the fixed-design setting,  $(t_i, Y_i)_{1 \leq i \leq n}$  are not identically distributed so that the definition of a “new observation”  $(T, Y)$  must be changed: Let  $T$  have a uniform distribution over  $\{t_1, \dots, t_n\}$  and  $Y = s(T) + \sigma(T)\epsilon$ , where  $\epsilon$  is independent from  $\epsilon_1, \dots, \epsilon_n$  and has the same distribution. Then,

$$\begin{aligned} \mathbb{E}_{(T,Y)} \left[ (\hat{s}(T) - Y)^2 \right] &= \frac{1}{n} \sum_{i=1}^n \mathbb{E}_{\epsilon} \left[ (s(t_i) + \sigma(t_i)\epsilon - \hat{s}(t_i))^2 \right] \\ &= \|s - \hat{s}\|_n^2 + \frac{1}{n} \sum_{i=1}^n \sigma(t_i)^2 . \end{aligned}$$

Hence, minimizing the CV estimator of  $\mathbb{E}_{(T,Y)} \left[ (\hat{s}_m(T) - Y)^2 \right]$  over  $m$  amounts to approximately minimize  $\|s - \hat{s}_m\|_n^2$ .

Even though the use of CV in a fixed-design setting is not usual, theoretical results detailed in Section 3.2.4 below show that CV actually leads to a good estimator of the quadratic risk  $\|s - \hat{s}_m\|_n^2$ . This fact is confirmed by all the experimental results of the paper.

### 3.2.2 Definition

Let us now formally define how CV is used for selecting some  $m \in \mathcal{M}_n(D)$  from data.

First, the hold-out estimator of the risk is defined as follows. Let  $I^{(t)} \subset \{1, \dots, n\}$ ,  $I^{(v)} = \{1, \dots, n\} \setminus I^{(t)}$ ,

$$P_n^{(t)} =: \frac{1}{\text{Card}(I^{(t)})} \sum_{i \in I^{(t)}} \delta_{(t_i, Y_i)} \quad \text{and} \quad P_n^{(v)} = \frac{1}{\text{Card}(I^{(v)})} \sum_{i \in I^{(v)}} \delta_{(t_i, Y_i)} .$$

The *hold-out estimator of the risk* of algorithm  $\mathcal{A} : P_n \mapsto \mathcal{A}(\cdot; P_n)$  is defined as

$$\widehat{R}_{ho}(\mathcal{A}, P_n, I^{(t)}) =: P_n^{(v)} \gamma \left( \mathcal{A} \left( P_n^{(t)} \right); \cdot \right) = \frac{1}{\text{Card}(I^{(v)})} \sum_{i \in I^{(v)}} \left( \mathcal{A}(t_i; P_n^{(t)}) - Y_i \right)^2 .$$

Cross-validation estimators of the risk of  $\mathcal{A}$  are then defined as the average of  $\widehat{R}_{ho}(\mathcal{A}, P_n, I^{(t)})$  over a family of training set indices  $I_1^{(t)}, \dots, I_B^{(t)}$ .

Leave-one-out, leave- $p$ -out and  $V$ -fold cross-validation are among the most classical examples of CV.

- *Leave-one-out* (Loo), often called *ordinary CV* [4, 49], consists in training with all the sample except one point, which is used for testing, repeatedly for each data point, that is  $I_j^{(t)} = \{1, \dots, n\} \setminus \{j\}$  for  $j = 1, \dots, n$ . The Loo estimator of the risk of  $\mathcal{A}$  is defined by

$$\widehat{R}_{Loo}(\mathcal{A}, P_n) =: \frac{1}{n} \sum_{j=1}^n \left[ \left( Y_j - \mathcal{A} \left( t_j; P_n^{(-j)} \right) \right)^2 \right],$$

where  $P_n^{(-j)} = (n-1)^{-1} \sum_{i \neq j} \delta_{(t_i, Y_i)}$ .

- *leave- $p$ -out* (Lpo) [20] is a generalization of the Loo with any  $p \in \{1, \dots, n-1\}$ . It consists in considering all subsets of  $\{1, \dots, n\}$  with cardinality  $n-p$  as training set indices  $I_j^{(t)}$ .

$$\widehat{R}_{Lpo(p)}(\mathcal{A}, P_n) =: \binom{n}{p}^{-1} \sum_{e \in \mathcal{E}_p} \left[ \frac{1}{p} \sum_{j \in e} \left[ \left( Y_j - \mathcal{A} \left( t_j; P_n^{(\bar{e})} \right) \right)^2 \right] \right], \quad (7)$$

where  $\mathcal{E}_p$  is the set of subsets of size  $p$  of  $\{1, \dots, n\}$  and for every  $e \in \mathcal{E}_p$ ,  $P_n^{(\bar{e})} =: (n-p)^{-1} \sum_{i \notin e} \delta_{(t_i, Y_i)}$ .

- *$V$ -fold cross-validation* (VFCV) has been introduced by Geisser [25, 26] has a computationally efficient alternative to Lpo and Loo. The idea is to partition first the data into  $V$  blocks, to use all the data but one block as a training sample, and to repeat the process  $V$  times; in other words, VFCV is a Loo on the blocks, so that its computational complexity is  $V$  times the complexity of  $\mathcal{A}$ . Formally, let  $B_1, \dots, B_V$  be a partition of  $\{1, \dots, n\}$  and  $P_n^{(B_k)} =: (n - \text{Card}(B_k))^{-1} \sum_{i \notin B_k} \delta_{(t_i, Y_i)}$  for every  $k \in \{1, \dots, V\}$ . The VFCV estimator of the risk of  $\mathcal{A}$  is defined by

$$\widehat{R}_{VF}(\mathcal{A}, P_n) =: \frac{1}{V} \sum_{k=1}^V \left[ \frac{1}{\text{Card}(B_k)} \sum_{j \in B_k} \left[ \left( Y_j - \mathcal{A} \left( t_j; P_n^{(B_k)} \right) \right)^2 \right] \right]. \quad (8)$$

The interested reader will find theoretical and experimental results on VFCV and the best way to perform it in practice in [8, 23] and references therein, in particular [20].

Given the Loo estimator for the risk of each algorithm  $\mathcal{A}$  among  $\{\text{ERM}(S_m; \cdot)\}_{m \in \mathcal{M}_n(D)}$ , the segmentation with  $D - 1$  breakpoints chosen by Loo is defined as follows.

**Algorithm 2.**

$$\hat{m}_{\text{Loo}}(D) := \text{Argmin}_{m|D_m=D} \hat{R}_{\text{Loo}}(\text{ERM}(S_m; \cdot), P_n)$$

The segmentations chosen by Lpo and VFCV are defined similarly and denoted respectively  $\hat{m}_{\text{Lpo}(p)}(D)$  and  $\hat{m}_{\text{VFCV}}(D)$ .

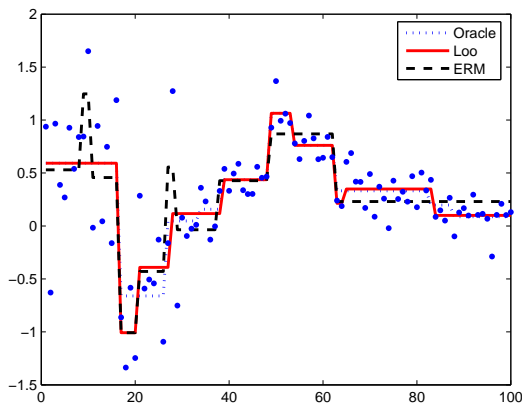


Figure 2: Comparison of  $\hat{m}_{\text{ERM}}(D)$  (dashed line),  $\hat{m}^*(D)$  (dotted line) and  $\hat{m}_{\text{Loo}}(D)$  (plain red line) with  $D = 10$ . Data are generated as on Figure 1, with a variable  $\sigma$ .

As illustrated by Figure 2,  $\hat{m}_{\text{Loo}}(D)$  is much closer to the oracle segmentation  $m^*(D)$  than  $\hat{m}_{\text{ERM}}(D)$  in the heteroscedastic setting used in the right panel of Figure 1. This improvement will be explained by theoretical results in Section 3.2.4 below.

### 3.2.3 Computational tractability

The computational complexity of  $\text{ERM}(S_m; P_n)$  is  $\mathcal{O}(n)$  since for every  $\lambda \in \Lambda_m$ , the value of  $\hat{s}_m(P_n)$  on  $I_\lambda$  is equal to the mean of  $\{Y_i\}_{t_i \in I_\lambda}$ . Therefore, a naive implementation of Lpo has a computational complexity  $\mathcal{O}\left(n \binom{n}{p}\right)$ , which can be intractable for large  $n$ , even when  $n = 1$ . In such cases,

only VFCV with a small  $V$  would work straightforwardly, since it has a computational complexity  $\mathcal{O}(nV)$ .

Nevertheless, closed-form formulas for the Lpo estimator of the risk have been derived in the density estimation [22, 21] and regression [23] frameworks. Some of these closed-form formulas apply to regressograms  $\widehat{s}_m$  as  $m \in \mathcal{M}_n$ . The following theorem gives closed-form expressions for  $\widehat{R}_{Lpo(p)}(\text{ERM}(S_m; \cdot), P_n)$  which only need a computational cost of order  $\mathcal{O}(n)$ .

**Theorem 1** (Corollary 3.3.2 in [23]). *Let  $m \in \mathcal{M}_n$ ,  $S_m$  and  $\widehat{s}_m = \text{ERM}(S_m; \cdot)$  be defined as in Section 2. For every  $(t_1, Y_1), \dots, (t_n, Y_n) \in \mathbb{R}^2$  and  $\lambda \in \Lambda_m$ , define*

$$S_{\lambda,1} = \sum_{j=1}^n Y_j \mathbb{1}_{\{t_j \in I_\lambda\}} \quad \text{and} \quad S_{\lambda,2} = \sum_{j=1}^n Y_j^2 \mathbb{1}_{\{t_j \in I_\lambda\}} .$$

Then, for every  $p \in \{1, \dots, n-1\}$ , the Lpo estimator of the risk of  $\widehat{s}_m$  defined by (7) is given by

$$\begin{aligned} \widehat{R}_{Lpo(p)}(m) = \sum_{\lambda \in \Lambda_m} \frac{1}{pN_\lambda} & \left[ \{S_{\lambda,2} (A_\lambda + C_\lambda(n_\lambda - 1)) - 2B_\lambda (S_{\lambda,1}^2 - S_{\lambda,2})\} \mathbb{1}_{\{n_\lambda \geq 2\}} \right. \\ & \left. + \{D_\lambda(n_\lambda - 2) [S_{\lambda,1}^2 - S_{\lambda,2}]\} \mathbb{1}_{\{n_\lambda \geq 3\}} + \{+\infty\} \mathbb{1}_{\{n_\lambda = 1\}} \right] , \end{aligned}$$

where for every  $\lambda \in \Lambda_m$ ,

$$\begin{aligned} n_\lambda &:= \text{Card}(\{i \mid t_i \in I_\lambda\}) & N_\lambda &:= 1 - \mathbb{1}_{\{p \geq n_\lambda\}} \binom{n - n_\lambda}{p - n_\lambda} / \binom{n}{p} \\ A_\lambda &= V_\lambda(0) - \frac{V_\lambda(1)}{n_\lambda} & B_\lambda &= \frac{A_\lambda}{n_\lambda - 1} \\ C_\lambda &= \frac{V_\lambda(-1)}{n_\lambda - 1} - \frac{V_\lambda(0)}{n_\lambda(n_\lambda - 1)} & D_\lambda &= \frac{(n_\lambda + 1)V_\lambda(0) - V_\lambda(1) - n_\lambda V_\lambda(-1)}{n_\lambda(n_\lambda - 1)(n_\lambda - 2)} \\ \text{and } \forall k \in \{-1, 0, 1\}, & V_\lambda(k) &= \sum_{r=\max\{1, (p-n_\lambda)\}}^{\min\{n_\lambda, (n-p)\}} r^k \frac{\binom{n-p}{r} \binom{p}{n_\lambda-r}}{\binom{n}{n_\lambda}} . \end{aligned}$$

An important practical consequence of Theorem 1 is that for every  $D, p$ ,  $\widehat{m}_{Lpo(p)}(D)$  can be computed with the same computational complexity as  $\widehat{m}_{\text{ERM}}(D)$ , that is  $\mathcal{O}(n^2)$ . Indeed, Theorem 1 shows that  $\widehat{R}_{Lpo(p)}(m)$  is a sum over  $\lambda \in \Lambda_m$  of terms depending only on  $\{Y_i\}_{t_i \in I_\lambda}$ , so that dynamic programming [15] can be used for computing the minimizer  $\widehat{m}_{Lpo(p)}(D)$  of



$\widehat{R}_{Lpo(p)}(m)$  over  $m \in \mathcal{M}_n$ . Therefore, *Lpo* and *Loo* are computationally tractable algorithms for finding the change-point detection when the number of breakpoints is given.

Dynamic programming also applies to  $\widehat{m}_{VFCV}$  with a computational complexity  $\mathcal{O}(Vn^2)$ , since each term appearing in  $\widehat{R}_{VFCV}(m)$  is the average over  $V$  quantities that must be computed, except when  $V = n$  since VFCV then becomes Loo. Since VFCV is mostly an approximation to Loo or Lpo,  $\widehat{m}_{Lpo(p)}(D)$  (with  $p$  to be chosen) will be preferred to  $\widehat{m}_{VFCV}(D)$  in the following.

### 3.2.4 Theoretical guarantees

In order to understand why CV indeed works for change-point detection with a given number of breakpoints, let us recall a straightforward consequence of Theorem 1 which is proved in details in [23, Lemma 7.2.1 and Proposition 7.2.3].

Using the same notations as in Lemma 1, for any  $m \in \mathcal{M}_n$ ,

$$\mathbb{E} \left[ \widehat{R}_{Lpo(p)}(m) \right] \approx \|s - s_m\|_n^2 + \frac{1}{n-p} \sum_{\lambda \in \Lambda_m} (\sigma_\lambda^r)^2 + \frac{1}{n} \sum_{i=1}^n \sigma(t_i)^2, \quad (9)$$

provided  $\min_{\lambda \in \Lambda_m} n_\lambda$  is large enough, in particular larger than  $p$ .

The comparison of (6) and (9) shows that Lpo yields an estimator of the risk  $\|s - \widehat{s}_m\|_n^2$  almost unbiased: The only difference is that the factor  $1/n$  in front of the variance term  $V(m)$  has been changed into  $1/(n-p)$ . Therefore, minimizing the Lpo estimator of the risk instead of the empirical risk allows to automatically take into account heteroscedasticity of data.

## 3.3 Simulation study

The goal of this section is to experimentally assess, for several values of  $p$ , the performance of Lpo in detecting a given number of changes in the mean of a heteroscedastic signal. This performance is also compared with that of empirical risk minimization.

### 3.3.1 Setting

The setting described in this section is used in all the experiments of Section 3.3; it will be extended in the following sections.

Data are generated according to (3) with  $n = 100$ . For every  $i$ ,  $t_i = i/n$  and  $\epsilon_i$  has a standard Gaussian distribution. The regression function

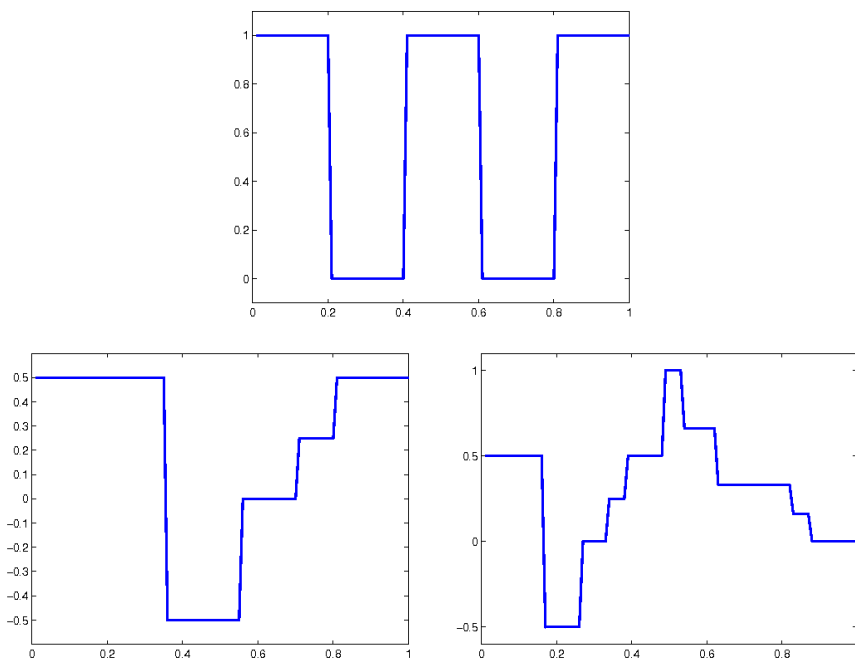


Figure 3: Regression functions  $s_1, s_2, s_3$ ;  $s_1$  and  $s_2$  are piecewise constant with 4 jumps;  $s_3$  is piecewise constant with 9 jumps.

$s$  is chosen among three piecewise constant functions  $s_1, s_2, s_3$  plotted on Figure 3. The noise-level function  $\sigma(\cdot)$  is chosen among the six following functions:

1. Homoscedastic noise:  $\sigma_c = 0.25 \mathbb{1}_{[0,1]}$ ,
2. Heteroscedastic piecewise constant noise:  $\sigma_{pc,1} = 0.2 \mathbb{1}_{[0,1/3]} + 0.05 \mathbb{1}_{[1/3,1]}$ ,  $\sigma_{pc,2} = 0.5 \mathbb{1}_{[0,1/3]} + 0.125 \mathbb{1}_{[1/3,1]}$  or  $\sigma_{pc,3} = 0.4 \mathbb{1}_{[0,1/3]} + 0.1 \mathbb{1}_{[1/3,1]}$ ,
3. Heteroscedastic sinusoidal noise:  $\sigma_{s,1} = 0.05 \sin(t\pi/4)$  or  $\sigma_{s,2} = 0.5 \sin(t\pi/4)$ .

All combinations between the regression functions  $(s_i)_{i=1,2,3}$  and the six noise-levels  $\sigma$ . have been considered, each time with  $N = 300$  independent samples. Results below only report a small part of the entire simulation study but intend to be representative of the main observed behaviour. A more complete report of the results, including other regression functions  $s$ , is given in Chapter 7 of the second authors' thesis [23].

### 3.3.2 Results: Comparison of segmentations for each dimension

The segmentations of each dimension  $D \in \{1, \dots, n\}$  obtained by empirical risk minimization (Algorithm 1) and Lpo (Algorithm 2) for several values of  $p$  are compared on Figure 4, through the expected values of the quadratic loss  $\mathbb{E} \left[ \|s - \widehat{s}_{\widehat{m}_A(D)}\|_n^2 \right]$ .

On the one hand, when data are homoscedastic (left of Figure 4), all methods yield similar performances for all dimensions up to twice the true dimension; Lpo is slightly better for larger dimensions. Therefore, unless the dimension is strongly overestimated (whatever the way  $D$  is chosen), all methods are equivalent with homoscedastic data.

On the other hand, when data are heteroscedastic (right of Figure 4), ERM yields significantly worse performance than Lpo for dimensions larger than half the true dimension. As explained in Sections 3.1 and 3.2.4,  $\widehat{m}_{\text{ERM}}(D)$  puts breakpoints inside pure noise for dimensions  $D$  smaller than the true dimension, whereas Lpo does not have this drawback. Therefore, whatever the choice of the dimension (except  $D \leq 4$ , that is for detecting the obvious jumps), Lpo should be preferred to empirical risk minimization as soon as data are heteroscedastic.

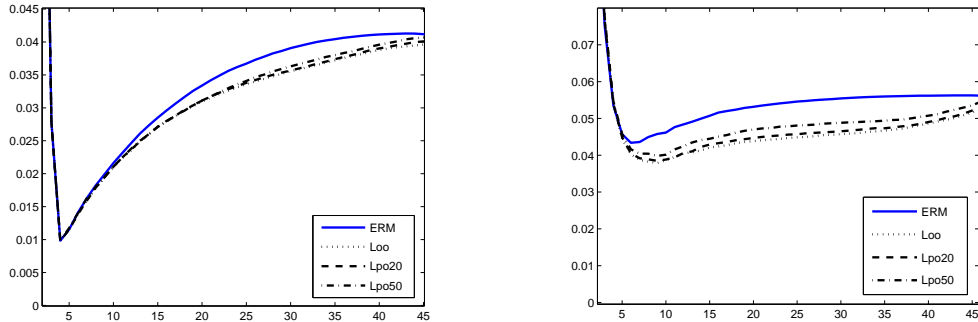


Figure 4:  $\mathbb{E} \left[ \left\| s - \widehat{s}_{\widehat{m}_{\mathcal{A}}(D)} \right\|_n^2 \right]$  as a function of  $D$  for  $\mathcal{A}$  among ‘ERM’ (empirical risk minimization), ‘Loo’ (Leave-one-out), ‘Lpo(20)’ (Lpo with  $p = 20$ ) and ‘Lpo(50)’ (Lpo with  $p = 50$ ). Left: homoscedastic data ( $s = s_2$ ,  $\sigma = \sigma_c$ ) Right: heteroscedastic data ( $s = s_3$ ,  $\sigma = \sigma_{pc,2}$ ).

### 3.3.3 Results: Comparison of the “best” segmentations

This section focuses on the segmentation obtained with the best possible choice of  $D$ , that is the one corresponding to the minimum of  $D \mapsto \left\| s - \widehat{s}_{\widehat{m}_{\mathcal{A}}(D)} \right\|_n^2$  which is plotted on Figure 4 for change-point detection methods  $\mathcal{A}$  among ERM, Loo and Lpo with  $p = 20$  and  $p = 50$ .

The performance of a method  $\mathcal{A}$  is measured by the expected ratio  $C_{\text{or}}(\mathcal{A})$  between the quadratic loss of the best segmentation in the sequence  $\{\widehat{m}_{\mathcal{A}}(D)\}_{D>0}$  and the quadratic loss of the oracle  $m^*$ :

$$C_{\text{or}}(\mathcal{A}) := \mathbb{E} \left[ \frac{\inf_{1 \leq D \leq n} \left\{ \left\| s - \widehat{s}_{\widehat{m}_{\mathcal{A}}(D)} \right\|_n^2 \right\}}{\inf_{m \in \mathcal{M}_n} \left\{ \left\| s - \widehat{s}_m \right\|_n^2 \right\}} \right].$$

Hence,  $C_{\text{or}}(\mathcal{A})$  measures what is lost compared to the oracle when selecting one segmentation  $\widehat{m}_{\mathcal{A}}(D)$  per dimension. Even if the choice of  $D$  is a real practical problem—which will be tackled in the next sections—,  $C_{\text{or}}(\mathcal{A})$  helps understand which is the best method for selecting a segmentation for each dimension.

Confirming the results of Section 3.3.2, Table 1 shows that all methods have similar performances when data are homoscedastic, with a small but not significant advantage for ERM compared to Lpo. On the contrary, when data are heteroscedastic, ERM performs significantly worse than Lpo for various regression and noise-level functions.

$s.$	$\sigma.$	ERM	Loo	Lpo(20)	Lpo(50)
2	c	<b>4.87</b> $\pm$ 0.62	<b>4.87</b> $\pm$ 0.62	<b>4.87</b> $\pm$ 0.62	4.99 $\pm$ 0.62
	pc,1	1.2 $\pm$ 0.081	1.2 $\pm$ 0.081	1.14 $\pm$ 0.056	<b>1.11</b> $\pm$ 0.044
	pc,2	<u>6.01</u> $\pm$ 0.53	4.14 $\pm$ 0.35	4.09 $\pm$ 0.35	<b>3.75</b> $\pm$ 0.34
3	c	<b>3.73</b> $\pm$ 0.1	3.8 $\pm$ 0.1	3.83 $\pm$ 0.1	<u>4</u> $\pm$ 0.11
	pc,1	<u>4.38</u> $\pm$ 0.23	<b>3.68</b> $\pm$ 0.22	3.69 $\pm$ 0.22	<b>3.68</b> $\pm$ 0.22
	pc,2	<u>6.86</u> $\pm$ 0.41	<b>6.02</b> $\pm$ 0.38	6.1 $\pm$ 0.4	6.41 $\pm$ 0.44

Table 1: Average performance  $C_{\text{or}}(\mathcal{A})$  for change-point detection methods  $\mathcal{A}$  among ERM, Loo and Lpo with  $p = 20$  and  $p = 50$ . Several regression functions  $s$  and noise-level functions  $\sigma$  have been considered, each time with  $N = 300$  independent samples. The **best performance** is bolded, while significantly worst values are underlined. Next to each value is indicated the corresponding empirical standard deviation divided by  $\sqrt{N}$ , measuring the uncertainty of the estimated performance.

## 4 Estimation of the number of breakpoints

In this section, the number of breakpoints is no longer assumed to be fixed or known *a priori*. The goal is to estimate it, as often needed when analyzing real data.

Two main methods are compared. On the one hand, a penalization method introduced by Birgé and Massart [17] was successfully used in previous papers on change-point detection [33, 35], assuming homoscedasticity of data. On the other hand, VFCV is proposed as an alternative to Birgé and Massart’s penalty (BM) when data may not be homoscedastic.

In order to make the comparison fair between BM and VFCV when focusing on the question of choosing the number of breakpoints, VFCV is used for choosing among the same segmentations as BM, that is  $\{\hat{m}_{\text{ERM}}(D)\}_{D \in \mathcal{D}}$ , where  $\mathcal{D} \subset \{1, \dots, n\}$  is the set of candidate dimensions. The combination of VFCV for choosing  $D$  with the new algorithms proposed in Section 3 will be studied in Section 5.

### 4.1 Birgé and Massart’s penalization

First, let us define precisely the penalization method proposed by Birgé and Massart [17] and successfully used for change-point detection in [33, 35].

**Algorithm 3** (Birgé and Massart [17]).

1.  $\forall m \in \mathcal{M}_n, \hat{s}_m := \text{ERM}(S_m, P_n)$  ,

2.  $\widehat{m}_{\text{BM}} := \text{Argmin}_{m \in \mathcal{M}_n, D_m \in \mathcal{D}} \{P_n \gamma(\widehat{s}_m) + \text{pen}_{\text{BM}}(m)\}$  ,  
 where for every  $m \in \mathcal{M}_n$ , the penalty  $\text{pen}_{\text{BM}}(m)$  only depends on  $S_m$  through its dimension:

$$\text{pen}_{\text{BM}}(m) = \text{pen}_{\text{BM}}(D_m) := \widehat{C} \left( c_1 \frac{D_m}{n} + c_2 \frac{D_m}{n} \log \left( \frac{n}{D_m} \right) \right) . \quad (10)$$

3.  $\widetilde{s}_{\text{BM}} := \widehat{s}_{\widehat{m}_{\text{BM}}}$ .

The positive constants  $c_1, c_2$  have been calibrated through simulations by Lebarbier [35], who advocates the choice  $c_1 = 5$  and  $c_2 = 2$ . The positive constant  $\widehat{C}$  has to be estimated from data; Lebarbier [35] and Lavielle [33] proposed to estimate it using Birgé and Massart's *slope heuristics* [18]; see also [9] for references and results about this heuristics.

All  $m \in \mathcal{M}_n(D)$  are penalized in the same way by  $\text{pen}_{\text{BM}}(m)$ , so that Algorithm 3 actually selects a segmentation among  $\{\widehat{m}_{\text{ERM}}(D)\}_{D \in \mathcal{D}}$ . Therefore, Algorithm 3 can be reformulated as follows, as noticed in Section 4.3 of [18].

**Algorithm 4** (Reformulation of Algorithm 3).

1.  $\forall D \in \{1, \dots, n\}$ ,  $\widehat{s}_{\widehat{m}_{\text{ERM}}(D)} := \text{ERM}(\widetilde{S}_D, P_n)$  where  $\widetilde{S}_D := \bigcup_{m \in \mathcal{M}_n(D)} S_m$  ,
2.  $\widehat{D}_{\text{BM}} := \text{Argmin}_{D \in \mathcal{D}} \{P_n \gamma(\widehat{s}_{\widehat{m}_{\text{ERM}}(D)}) + \text{pen}_{\text{BM}}(D)\}$  where  $\text{pen}_{\text{BM}}(D)$  is defined by (10),
3.  $\widetilde{s}_{\text{BM}} := \widehat{s}_{\widehat{m}_{\text{ERM}}(\widehat{D}_{\text{BM}})}$  .

In the following, 'BM' denotes Algorithm 4 and

$$\text{crit}_{\text{BM}}(D) := P_n \gamma(\widehat{s}_{\widehat{m}_{\text{ERM}}(D)}) + \text{pen}_{\text{BM}}(D)$$

is called the BM criterion.

Algorithm 4 clarifies the reason why  $\text{pen}_{\text{BM}}$  must be larger than Mallows'  $C_p$  penalty. Indeed, for every  $m \in \mathcal{M}_n$ , Lemma 1 shows that when data are homoscedastic,  $P_n \gamma(\widehat{s}_m) + \text{pen}(m)$  is an unbiased estimator of  $\|s - \widehat{s}_m\|_n^2$  when  $\text{pen}(m) = 2\sigma^2 D_m n^{-1}$ , that is Mallows'  $C_p$  penalty. When  $\text{Card}(\mathcal{M}_n)$  is at most polynomial in  $n$ , Mallows' penalty leads to an efficient model selection procedure, as proved in several frameworks [47, 36, 11], meaning

that Mallows’ penalty is an adequate measure of the “complexity” of any vector space  $S_m$  of dimension  $D_m$ .

On the contrary, in the change-point detection framework,  $\text{Card}(\mathcal{M}_n)$  is exponential in  $n$ . Algorithm 4 formulation points out that Birgé and Massart’s penalization amounts to choose the penalty  $\text{pen}_{\text{BM}}(D)$  such that for every  $D$ ,  $\text{crit}_{\text{BM}}(D)$  is an unbiased estimator of  $\|s - \hat{s}_{\hat{m}(D)}\|_n^2$ , where  $\hat{s}_{\hat{m}(D)}$  is the empirical risk minimizer of  $\tilde{S}_D$ . Hence,  $\text{pen}_{\text{BM}}(D)$  measures the “complexity” of  $\tilde{S}_D$ , which is much more complex than a vector space of dimension  $D$ . Therefore,  $\text{pen}_{\text{BM}}$  should naturally be larger than Mallows’  $C_p$ , as confirmed by the results of Birgé and Massart [18] on minimal penalties for exponential collections of models.

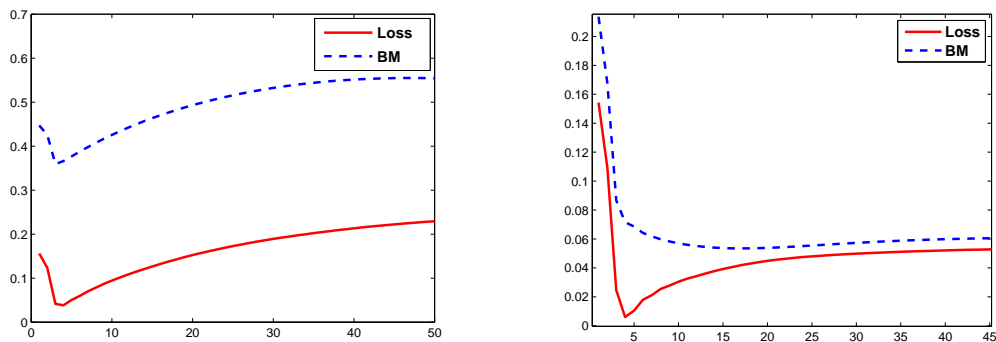


Figure 5: Comparison of  $\|s - \hat{s}_{\hat{m}(D)}\|_n^2$  (‘Loss’) and  $\text{crit}_{\text{BM}}(D)$  (‘BM’). Data are generated as explained in Section 3.3.1. Left:  $s = s_2$ ,  $\sigma = \sigma_c$  (homoscedastic). Right:  $s = s_2$ ,  $\sigma = \sigma_{pc,2}$  (heteroscedastic).

Simulation experiments (Figure 5 left) support the fact that  $\text{crit}_{\text{BM}}(D)$  is an unbiased estimator of  $\|s - \hat{s}_{\hat{m}(D)}\|_n^2$  for every  $D$  (up to an additive constant) when data are homoscedastic. However, when data are heteroscedastic, theoretical results proved by Birgé and Massart [17, 18] no longer apply, and simulations show that  $\text{crit}_{\text{BM}}(D)$  does not always estimate  $\|s - \hat{s}_{\hat{m}(D)}\|_n^2$  well, at least when  $\hat{C}$  is estimated as in [33, 35] (Figure 5 right). This result is consistent with Lemma 1, as well as the result proved by the first author [7] that penalties proportional to  $D_m$  can be suboptimal for model selection among a polynomial collection of models when data are heteroscedastic.

Therefore,  $\text{pen}_{\text{BM}}(D)$  is not an adequate complexity measure of  $\tilde{S}_D$  in

general when data are heteroscedastic and another complexity measure is required.

## 4.2 Cross-validation

As shown in Section 3.2.2, CV can be used for estimating the quadratic loss  $\|s - \mathcal{A}(P_n)\|_n^2$  of any algorithm. In particular, CV was used successfully in Section 3 for estimating the quadratic risk of  $\text{ERM}(S_m, \cdot)$  for all segmentations  $m \in \mathcal{M}_n(D)$  with a given number of breakpoints  $D - 1$  (Algorithm 2), even when data are heteroscedastic.

Therefore, CV methods are natural candidates for fixing BM's failure observed on the right panel of Figure 5. The proposed algorithm—with VFCV—is the following.

### Algorithm 5.

1.  $\forall D \in \mathcal{D}, \hat{s}_{\widehat{m}_{\text{ERM}}(D)} := \text{ERM}(\widetilde{S}_D, P_n)$  ,
2.  $\widehat{D}_{\text{VF}} := \text{Argmin}_{D \in \mathcal{D}} \{\text{crit}_{\text{VF}}(D)\}$  where  $\text{crit}_{\text{VF}}(D) := \widehat{R}_{\text{VF}}(\text{ERM}(\widetilde{S}_D, \cdot), \cdot)$  and  $\widehat{R}_{\text{VF}}$  is defined by (8).

*Note that  $\widehat{R}_{\text{VF}}$  is not well-defined if models  $m \in \mathcal{M}_n$  such that  $\exists k, \lambda \in \Lambda_m$  with  $\{t_i \in I_\lambda \text{ and } i \notin B_k\} = \emptyset$  are not removed from  $\mathcal{M}_n$ . To this aim, the blocks  $B_k$  are chosen such that two consecutive design points belong to the same block, and models  $S_m$  such that  $\min_{\lambda \in \Lambda_m} \text{Card}\{t_i \in I_\lambda\} < 2$  are removed from  $\widetilde{S}_D$ .*

Similar algorithms can be defined with Loo and Lpo instead of VFCV. The interest of VFCV is its reasonably small computational cost—taking  $V \leq 10$  for instance—, since no closed-form formulas exist for CV estimators of the risk of  $\text{ERM}(\widetilde{S}_D, P_n)$ .

## 4.3 Simulation results

A simulation experiment was performed in the setting presented in Section 3.3.1. A representative picture of the results is given by Figure 6 and by Table 2.

As illustrated by Figure 6, VFCV indeed correctly estimates the risk of empirical risk minimizers over  $\widetilde{S}_D$  for every  $D$  and for both homoscedastic and heteroscedastic data. Hence, the actual complexity of  $\widetilde{S}_D$  is taken into account by the criterion  $\text{crit}_{\text{VF}}(D)$ .



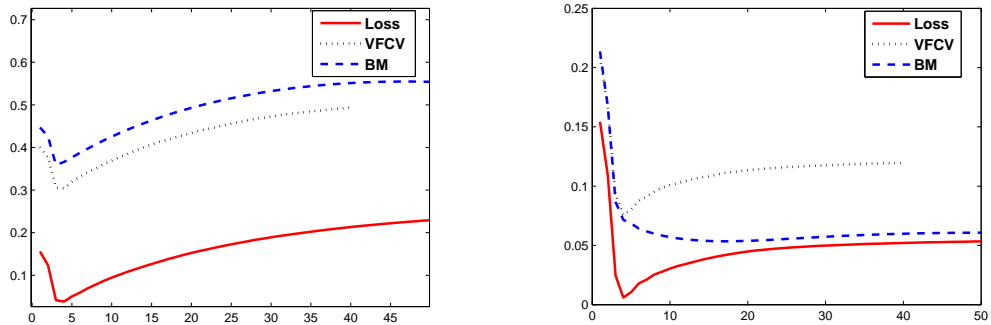


Figure 6: Comparison of  $\|s - \widehat{s}_{\widehat{m}(D)}\|_n^2$  ('Loss'),  $\text{crit}_{\text{VF}}(D)$  ('VFCV') and  $\text{crit}_{\text{BM}}(D)$  ('BM'). Data are generated as explained in Section 3.3.1. Left:  $s = s_2$ ,  $\sigma = \sigma_c$  (homoscedastic). Right:  $s = s_2$ ,  $\sigma = \sigma_{pc,2}$  (heteroscedastic).

Subsequently, VFCV yields a much smaller loss  $\|s - \widehat{s}_{\widehat{m}_{\text{ERM}}(\widehat{D})}\|_n^2$  than BM when data are heteroscedastic (Table 2). When data are homoscedastic, VFCV and BM provide similar values for the risk with an insignificant advantage for BM; this is not surprising since BM makes use of the knowledge that data are homoscedastic. Moreover, it has been proved to be optimal in this setting [17].

Therefore, VFCV appears to be a reliable alternative to BM when no *a priori* knowledge guarantees that data are homoscedastic.

## 5 New change-point detection algorithms via cross-validation

Sections 3 and 4 showed that when data are heteroscedastic, CV can be used successfully instead of penalized criteria for detecting breakpoints given their number as well as for estimating the number of breakpoints. Nevertheless, in Section 4, the segmentations compared by CV were obtained by empirical risk minimization so that they may be suboptimal according to the results of Section 3.

The next step for obtaining reliable change-point detection algorithms for heteroscedastic data is to combine the two ideas, that is use CV twice. The goal of the present section is to properly define such algorithms (with various kinds of CV) and to assess their performances.

$s.$	$\sigma.$	Oracle	VFCV ( $V = 5$ )	BM
1	c	$2.19 \pm 0.28$	$3.35 \pm 0.35$	<b><math>2.96 \pm 0.36</math></b>
	pc,3	$4.04 \pm 0.57$	<b><math>7.33 \pm 0.9</math></b>	<u><math>38.5 \pm 2.8</math></u>
	s,2	$3.83 \pm 0.36$	<b><math>4.92 \pm 0.4</math></b>	<u><math>7.86 \pm 1.3</math></u>
2	c	$4.17 \pm 0.24$	$5.65 \pm 0.33$	<b><math>5.52 \pm 0.37</math></b>
	pc,3	$5.97 \pm 0.49$	<b><math>12.5 \pm 1.2</math></b>	<u><math>59.4 \pm 5</math></u>
	s,2	$5.82 \pm 0.41$	<b><math>8.02 \pm 0.51</math></b>	<u><math>10.3 \pm 0.76</math></u>

Table 2: Average of the quadratic loss  $\|s - \hat{s}_{\hat{m}_{\text{ERM}}(D)}\|_n^2$  for the oracle dimension  $D^* := \text{Argmin}_{D \in \mathcal{D}} \left\{ \|s - \hat{s}_{\hat{m}_{\text{ERM}}(D)}\|_n^2 \right\}$ , for  $D = \hat{D}_{\text{VF}}$  with  $V = 5$  and for  $D = \hat{D}_{\text{BM}}$ . Several regression functions  $s$  and noise-level functions  $\sigma$  have been considered, each time with  $N = 300$  independent samples. The **best performance** is bolded, while significantly worst values are underlined. Next to each value is indicated the corresponding empirical standard deviation divided by  $\sqrt{N}$ , measuring the uncertainty of the estimated performance.

## 5.1 Definition of a family of change-point detection algorithms

The general strategy used in this article for change-point detection relies on two steps: First, detect where  $D - 1$  breakpoints should be located for every  $D \in \mathcal{D}$ ; second, estimate the number  $D - 1$  of breakpoints. This strategy can be summarized with the following algorithm:

**Algorithm 6** (General two-steps change-point detection algorithm).

1.  $\forall D \in \mathcal{D}$ ,  $\mathcal{A}_D(P_n) := \hat{s}_{\hat{m}(D)} = \text{Argmin}_{m \in \mathcal{M}_n(D)} \{ \text{crit}_1(S_m, P_n) \}$  where for every model  $S$ ,  $\text{crit}_1(S, P_n) \in \mathbb{R}$  estimates  $\|s - \text{ERM}(S, P_n)\|_n^2$  and  $\hat{s}_m = \text{ERM}(S_m, P_n)$  is defined as in Section 3.1.
2.  $\hat{D} = \text{Argmin}_{D \in \mathcal{D}} \{ \text{crit}_2(\mathcal{A}_D, P_n) \}$ , where for every algorithm  $\mathcal{A}_D$ ,  $\text{crit}_2(\mathcal{A}_D, P_n) \in \mathbb{R}$  estimates  $\|s - \mathcal{A}_D(P_n)\|_n^2$ .
3. Output: the segmentation  $\hat{m}(\hat{D})$  and the corresponding estimator  $\hat{s}_{\hat{m}(\hat{D})}$  of  $s$ .

Let us now detail which are the candidate criteria  $\text{crit}_1$  and  $\text{crit}_2$  for being used in Algorithm 6. For the first step:

- The ideal criterion ('Id') is  $\text{crit}_{1,Id}(S, P_n) := \|s - \text{ERM}(S, P_n)\|_n^2$ .
- The empirical risk ('Emp') is

$$\text{crit}_{1,\text{Emp}}(S, P_n) := P_n \gamma(\text{ERM}(S, P_n))$$

- The Leave- $p$ -out estimator of the risk ('Lpo( $p$ )') is, for every  $p \in \{1, \dots, n-1\}$ ,

$$\text{crit}_{1,\text{Lpo}}(S, P_n, p) := \widehat{R}_{\text{Lpo}(p)}(\text{ERM}(S; \cdot), P_n)$$

As in Section 3, Lpo(1) will be noted Loo. Note that the VFCV estimator of the risk  $\widehat{R}_{\text{VF}}$  could also be used as  $\text{crit}_1$ ; it will not be considered in the following because it is computationally more expensive and more variable than Lpo.

For the second step:

- The ideal criterion ('Id') is  $\text{crit}_{2,Id}(\mathcal{A}_D, P_n) := \|s - \mathcal{A}_D(P_n)\|_n^2$ .
- Birgé and Massart's penalization criterion ('BM') is

$$\text{crit}_{2,\text{BM}}(\mathcal{A}_D, P_n) := P_n \gamma(\mathcal{A}_D(P_n)) + \text{pen}_{\text{BM}}(D) ,$$

where  $\text{pen}_{\text{BM}}(D)$  is defined by (10) with  $c_1 = 5$ ,  $c_2 = 2$  and  $\widehat{C}$  is chosen by the slope heuristics [18, 9].

- The  $V$ -fold cross-validation estimator of the risk ('VF( $V$ )') is, for every  $V \in \{1, \dots, n\}$ ,

$$\text{crit}_{2,\text{VF}(V)}(\mathcal{A}_D, P_n) := \widehat{R}_{\text{VF}(V)}(\mathcal{A}_d, P_n) ,$$

where  $\widehat{R}_{\text{VF}}$  is defined by (8) and the blocks  $B_1, \dots, B_V$  are chosen as in Algorithm 5.

REMARK: For  $\text{crit}_2$ , definitions using Lpo could be formally considered. However, they are not investigated here because of their prohibitive computational cost.

In the following, the notation  $\llbracket A, B \rrbracket$  will be used as a shortcut for "Algorithm 6 with  $\text{crit}_{1,A}$  and  $\text{crit}_{2,B}$ ". For instance, BM coincides with  $\llbracket \text{Emp}, \text{BM} \rrbracket$ ; Algorithms  $\llbracket A, Id \rrbracket$  were compared for different  $A$  in Section 3; Algorithms  $\llbracket \text{Emp}, B \rrbracket$  were compared for several  $B$  in Section 4.

## 5.2 Simulation study

A simulation experiment was performed for comparing algorithms  $\{\llbracket \mathcal{A}, \text{VF}(5) \rrbracket\}_A$  in the setting presented in Section 3.3.1. A representative picture of the results is given by Table 3. The performance of each competing algorithm  $\mathcal{A}$  is measured by the expected ratio  $C_{\text{or}}(\mathcal{A})$  between the quadratic loss of  $\mathcal{A}(P_n)$  and that of the oracle  $m^*(D^*)$ , that is

$$C_{\text{or}}(\mathcal{A}) =: \mathbb{E} \left[ \frac{\|s - \mathcal{A}(P_n)\|_n^2}{\inf_{m \in \mathcal{M}_n} \left\{ \|s - \widehat{s}_m(P_n)\|_n^2 \right\}} \right],$$

the expectation being evaluated by averaging over  $N = 500$  independent samples.

$s.$	$\sigma_{c.}$	$\llbracket \text{Emp}, \text{VF}(5) \rrbracket$	$\llbracket \text{Loo}, \text{VF}(5) \rrbracket$	$\llbracket \text{Lpo}(20), \text{VF}(5) \rrbracket$	$\llbracket \text{Lpo}(50), \text{VF}(5) \rrbracket$
1	c	$2.85 \pm 0.2$	<b><math>2.78 \pm 0.18</math></b>	<u><math>3.42 \pm 0.35</math></u>	$2.79 \pm 0.18$
	pc,1	$4.81 \pm 0.47$	$4.56 \pm 0.46$	$4.62 \pm 0.49$	<b><math>4.54 \pm 0.47</math></b>
	pc,2	$6.45 \pm 0.55$	$6.51 \pm 0.55$	$6.43 \pm 0.55$	<b><math>6.41 \pm 0.54</math></b>
	s,1	<u><math>7.43 \pm 0.74</math></u>	<b><math>5.73 \pm 0.37</math></b>	$6.03 \pm 0.41$	$5.75 \pm 0.41$
2	c	<b><math>5.85 \pm 0.33</math></b>	$6.19 \pm 0.39$	$6.1 \pm 0.36$	$6.26 \pm 0.38$
	pc,1	$8 \pm 1.3$	$7.58 \pm 1.2$	$7.31 \pm 1.1$	<b><math>7.27 \pm 1.2</math></b>
	pc,2	<u><math>13.8 \pm 1.4</math></u>	$12.5 \pm 1.5$	$11 \pm 1.2$	<b><math>10.6 \pm 0.88</math></b>
	s,1	$5.77 \pm 0.56$	$5.89 \pm 0.45$	$6.01 \pm 0.55$	<b><math>5.23 \pm 0.4</math></b>
3	c	<b><math>5.05 \pm 0.13</math></b>	$5.22 \pm 0.12$	$5.26 \pm 0.13$	<u><math>5.58 \pm 0.12</math></u>
	pc,1	<u><math>7.5 \pm 0.34</math></u>	$6.51 \pm 0.31$	<b><math>6.47 \pm 0.3</math></b>	$6.86 \pm 0.4$
	pc,2	<u><math>9.7 \pm 0.35</math></u>	<b><math>8.65 \pm 0.34</math></b>	$8.85 \pm 0.37$	<u><math>9.46 \pm 0.37</math></u>
	s,1	<u><math>5.5 \pm 0.26</math></u>	<b><math>4.98 \pm 0.21</math></b>	$5.12 \pm 0.21$	$4.99 \pm 0.22$

Table 3: Ratio of the average loss obtained for model provided by  $1^*2\text{VF}_5$  over that of the pathwise oracle (minimizer of  $\|s - \widehat{s}_m\|_n^2$ ) for each regression function and each noise.  $N = 500$  repetitions have been made. Next to each value is indicated the corresponding empirical standard deviation.

On the one hand, when data are homoscedastic, nearly no significant difference arises between the competing algorithms. On the other hand, when data are heteroscedastic, the use of Lpo for choosing the best segmentation of each dimension yields significantly better performances than empirical risk minimization. Hence, the results of Section 3 are confirmed when using VFCV for choosing the dimension, instead of the ‘Id’ criterion that cannot be used in practice.

In addition, depending on the simulation setting, significantly better performances are obtained for  $Lpo(p)$  with  $p = 1$  or  $p = 50$ . Therefore, an efficient automatic choice of  $p$  would be of great interest in practice. As noticed in previous works by the authors', the choice of  $p$  for  $Lpo(p)$  is related to overpenalization [6, 21, 23], which is itself required for large collections of models or small signal-to-noise ratios [6, Section 7.3].

## 6 Application to CGH microarray data

In this section, the new change-point detection algorithms proposed in the paper are applied to CGH microarray data.

### 6.1 Biological context

The purpose of CGH microarray experiments is to detect and map chromosomal aberrations. For instance, a piece of chromosome can be *amplified*, that is appear several times more than usual, or *deleted*. Such aberrations are often related to cancer disease.

Roughly, CGH profiles give the log-ratio of the DNA copy number along the chromosomes, compared to a reference DNA sequence. The interested reader will find more details about the biological context of CGH data in [40–42] for instance.

The goal of CGH data analysis is to detect changes in the mean of a signal (the log-ratio of copy numbers), and to estimate the mean in each segment. Hence, change-point detection algorithms are needed.

Moreover, assuming that CGH data are homoscedastic can be over-optimistic. Indeed, changes in the chemical composition of the sequence between normal and altered regions are known to induce changes in the variance of the observed signal. Therefore, algorithms robust to heteroscedasticity, such as the ones proposed in Section 5, should yield better results—in terms of detecting changes of copy number—than algorithms assuming homoscedasticity.

The data set considered in this section is based on the Bt474 cell lines [41], which denote epithelial cells obtained from human breast cancer tumors on a 60-year-old woman. A test genome of Bt474 cell lines is compared to a normal reference male genome. Even though several chromosomes are studied in these cell lines, this section focuses on chromosomes 1 and 9. Whereas chromosome 1 exhibits a putative heterogenous variance along the

CGH profile, chromosome 9 is likely to meet the homoscedasticity assumption. For each chromosome, log-ratios of copy numbers have been measured at 119 (respectively 93) locations.

## 6.2 Change-point detection algorithms used in the CGH literature

Before applying Algorithm 6 to the analysis of Bt474 CGH data, let us recall the definition of two change-point detection algorithms previously used in the literature for analyzing the same data.

The first algorithm is a simplified version of BM (Algorithm 4), denoted by ‘BMsimple’ [41]. The penalty  $\text{pen}_{\text{BM}}(D)$  is replaced by  $\widehat{C}'D$ , the constant  $\widehat{C}'$  being estimated from data by the slope heuristics algorithm [33, 35]. Note that  $\text{pen}_{\text{BM}}(D)$  can be written as  $\beta D - \gamma D \log(D)$  for some positive data-dependent  $\beta, \gamma$ . Hence, the only difference between  $\widehat{C}'D$  and  $\text{pen}_{\text{BM}}(D)$  occurs for large  $D$ , that is when the term  $\gamma D \log(D)$  actually matters in  $\text{pen}_{\text{BM}}(D)$ .

The second algorithm—denoted by ‘MV’—aims at detecting changes in either the mean or the variance, that is breakpoints for  $(s, \sigma)$  [41]. The selected model is defined as the minimizer over  $m \in \mathcal{M}_n$  of

$$\text{crit}_{\text{MV}}(m) =: \sum_{\lambda \in \Lambda_m} n_\lambda \log \left( \frac{1}{n_\lambda} \sum_{t_i \in I_\lambda} (Y_i - \widehat{s}_m(t_i; P_n))^2 \right) + \beta D_m ,$$

where  $n_\lambda = \text{Card} \{t_i \in I_\lambda\}$ .

## 6.3 Results

Results obtained with BMsimple, MV,  $\llbracket \text{Emp}, \text{VF}(5) \rrbracket$  and  $\llbracket \text{Lpo}(20), \text{VF}(5) \rrbracket$  on the Bt474 data set are reported on Figure 7 (chromosome 9) and 8 (chromosome 1).

For chromosome 9, BMsimple and MV yield (almost) the same segmentation, so that the homoscedasticity assumption is certainly not much violated. As expected,  $\llbracket \text{Emp}, \text{VF}(5) \rrbracket$  and  $\llbracket \text{Lpo}(20), \text{VF}(5) \rrbracket$  also yield very similar segmentations, which confirms the reliability of these algorithms for homoscedastic signal. (See Section 7.6 in [23] for more details.)

The picture is quite different for chromosome 1. Indeed, as shown by Figure 8, BMsimple selects a segmentation with 7 breakpoints, whereas MV selects a segmentation with only one breakpoint. The major difference

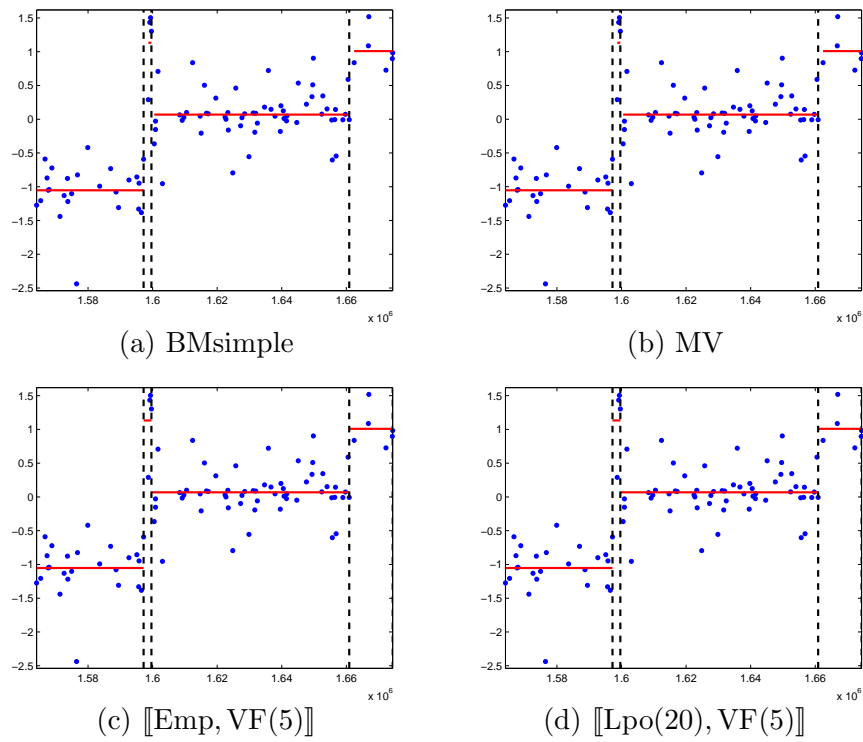


Figure 7: Change-points locations along chromosome 9. The mean on each homogeneous region is indicated by plain horizontal lines.

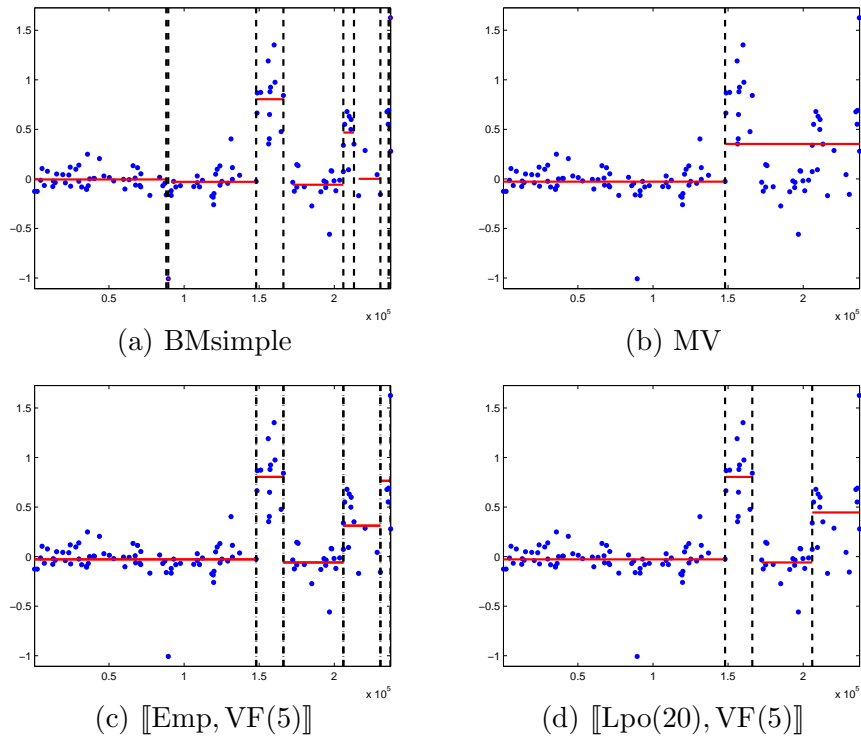


Figure 8: Change-points locations along Chromosome 1. The mean on each homogeneous region is indicated by plain horizontal lines.



between BMsimple and MV supports at least the idea that this data set is likely heteroscedastic.

Nevertheless, none of the segmentations chosen by BMsimple and MV are entirely satisfactory: The former relies on an assumption which is certainly violated; the latter may use a change in the estimated variance for explaining several changes in the mean.

CV-based algorithms  $\llbracket \text{Emp}, \text{VF}(5) \rrbracket$  and  $\llbracket \text{Lpo}(20), \text{VF}(5) \rrbracket$  yield two other segmentations, with an intermediate number of breakpoints, respectively 4 and 3. In view of the simulation experiments of the previous sections, the segmentation obtained via  $\llbracket \text{Lpo}(20), \text{VF}(5) \rrbracket$  should be the most reliable one since data are heteroscedastic. Therefore, Figure 8 can be interpreted as follows: The noise-level is small in the first part of chromosome 1, then higher, but not as high as estimated by MV. In particular, the copy number changes twice inside the second part of chromosome 1 (as defined by the segmentation obtained with MV), indicating that two regions of chromosome 1 are putative amplified.

Note however that choosing among the segmentations obtained with  $\llbracket \text{Emp}, \text{VF}(5) \rrbracket$  and  $\llbracket \text{Lpo}(20), \text{VF}(5) \rrbracket$  is not an easy task without additional data. A definitive answer would need further biological experiments.

## 7 Conclusion

### 7.1 Summary of the results

Cross-validation (CV) algorithms have been shown to be reliable for detecting changes in the mean of a signal, even when the signal is heteroscedastic.

First, when the number of breakpoints is given, the widely used empirical risk minimization has been proved to fail for some heteroscedasticity problems, from both theoretical and experimental aspects. On the contrary, the Leave- $p$ -out (Lpo) is robust to heteroscedasticity while being computationally efficient thanks to closed-form formulas given in Section 3.2.3 (Theorem 1).

Second, for choosing the number of breakpoints, the commonly used penalization proposed by Birgé and Massart, derived from a homoscedastic framework, cannot be applied as such to heteroscedastic data.  $V$ -fold cross-validation (VFCV) turns out to be a reliable alternative— both with homoscedastic and heteroscedastic data—, leading to much better segmentations in terms of quadratic risk when the variance is heterogenous. Besides unlike usual deterministic penalized criteria, VFCV efficiently chooses

among segmentations obtained by either Lpo or empirical risk minimization, without any specific change in the procedure.

To conclude, the combination of Lpo (for choosing a segmentation for each possible number of breakpoints) and VFCV seems to be the most reliable method for detecting changes in the mean of a signal which is not *a priori* known to be homoscedastic. This algorithm is computationally tractable, since its computational complexity is of order  $\mathcal{O}(Vn^2)$ , which is similar to most of change-point detection algorithms for small values of  $V$ ; in our simulation experiments,  $V = 5$  is shown to yield a reasonable good statistical performance. When applied to real data (CGH profiles in Section 6), the proposed algorithm turns out to be quite useful and effective, for a data set on which existing methods highly disagree because of heteroscedasticity.

## 7.2 Prospects

From the simulation experiment presented here, it appears that choosing the parameter  $p$  of  $\text{Lpo}(p)$  is intricate because of its dependence on the regression function and the noise structure. It therefore requires some further investigations. While choosing  $p > 1$  can improve upon  $p = 1$ ,  $\text{Lpo}(p)$  with a too large  $p$  can conversely lead to poor statistical performances. Any data-driven method for choosing  $p$  would be of great interest in practice.

Other resampling algorithms than CV can also be considered for change-point detection, for instance  $V$ -fold and Rademacher penalties [6, 7]. Using these approaches inside Algorithm 6 leads to a broad family of resampling-based change-point detection algorithms, which deserves further theoretical and experimental works.

## References

- [1] F. Abramovich, Y. Benjamini, D. Donoho, and I. Johnstone. Adapting to Unknown Sparsity by controlling the False Discovery Rate. *The Annals of Statistics*, 34(2):584–653, 2006.
- [2] H. Akaike. Statistical predictor identification. *Ann. Inst. Statist. Math.*, 22:203–217, 1969.
- [3] Hirotugu Akaike. Information theory and an extension of the maximum likelihood principle. In *Second International Symposium on Information Theory (Tsahkadsor, 1971)*, pages 267–281. Akadémiai Kiadó, Budapest, 1973.

- [4] David M. Allen. The relationship between variable selection and data augmentation and a method for prediction. *Technometrics*, 16:125–127, 1974.
- [5] Sylvain Arlot. *Resampling and Model Selection*. PhD thesis, University Paris-Sud 11, December 2007. oai:tel.archives-ouvertes.fr:tel-00198803\_v1.
- [6] Sylvain Arlot. Model selection by resampling penalization, 2008. hal-00262478.
- [7] Sylvain Arlot. Suboptimality of penalties proportional to the dimension for model selection in heteroscedastic regression, December 2008. arXiv:0812.3141.
- [8] Sylvain Arlot.  $V$ -fold cross-validation improved:  $V$ -fold penalization, February 2008. arXiv:0802.0566v2.
- [9] Sylvain Arlot and Pascal Massart. Data-driven calibration of penalties for least-squares regression. *J. Mach. Learn. Res.*, 2009. To appear. arXiv:0802.0837v4.
- [10] Yannick Baraud. Model selection for regression on a fixed design. *Probab. Theory Related Fields*, 117(4):467–493, 2000.
- [11] Yannick Baraud. Model selection for regression on a random design. *ESAIM Probab. Statist.*, 6:127–146 (electronic), 2002.
- [12] A. Barron, L. Birgé, and P. Massart. Risk bounds for model selection via penalization. *Probab. Theory and Relat. Fields*, 113:301–413, 1999.
- [13] P. Bartlett, S. Boucheron, and G. Lugosi. Model selection and error estimation. *Machine Learning*, 48(1–3):85–113, 2002.
- [14] M. Basseville and N. Nikiforov. *The Detection of Abrupt Changes - Theory and Applications*. Prentice-Hall: Information and System Sciences Series, 1993.
- [15] R. E. Bellman and S. E. Dreyfus. *Applied Dynamic Programming*. Princeton, 1962.
- [16] L. Birgé and P. Massart. From model selection to adaptive estimation. In D. Pollard, E. Torgensen, and G. Yang, editors, *In Festschrift for Lucien Le Cam: Research Papers in Probability and Statistics*, pages 55–87. Springer-Verlag, New York, 1997.

- [17] L. Birgé and P. Massart. Gaussian model selection. *J. European Math. Soc.*, 3(3):203–268, 2001.
- [18] Lucien Birgé and Pascal Massart. Minimal penalties for Gaussian model selection. *Probab. Theory Related Fields*, 138(1-2):33–73, 2007.
- [19] B. Brodsky and B. Darkhovsky. *Methods in Change-point problems*. Kluwer Academic Publishers, Dordrecht, The Netherlands, 1993.
- [20] P. Burman. Comparative study of Ordinary Cross-Validation, v-Fold Cross-Validation and the repeated Learning-Testing Methods. *Biometrika*, 76(3):503–514, 1989.
- [21] A. Celisse. Density estimation via cross-validation: Model selection point of view. Technical report, arXiv, 2008.
- [22] A. Celisse and S. Robin. Nonparametric density estimation by exact leave-p-out cross-validation. *Computational Statistics and Data Analysis*, 52(5):2350–2368, 2008.
- [23] Alain Celisse. *Model selection via cross-validation in density estimation, regression and change-points detection*. PhD thesis, University Paris-Sud 11, December 2008. oai:tel.archives-ouvertes.fr:tel-00346320\_v1.
- [24] B. Efron. How biased is the Apparent Error Rate of a Prediction Rule? *J. Amer. Statist. Assoc.*, 81(394):461–470, 1986.
- [25] S. Geisser. A predictive approach to the random effect model. *Biometrika*, 61(1):101–107, 1974.
- [26] Seymour Geisser. The predictive sample reuse method with applications. *J. Amer. Statist. Assoc.*, 70:320–328, 1975.
- [27] Xavier Gendre. Simultaneous estimation of the mean and the variance in heteroscedastic gaussian regression, 2008.
- [28] P. Hall. *The Bootstrap and Edgeworth Expansion*. Springer-Verlag, New York, 1992.
- [29] P. Hall and E. Mammen. On general resampling algorithms and their performance in distribution estimation. *The Annals of Statistics*, 22(4):2011–2030, 1994.

- [30] M. Kearns, Y. Mansour, A. Y. Ng, and D. Ron. An Experimental and Theoretical Comparison of Model Selection Methods. *Machine Learning*, 27:7–50, 1997.
- [31] P. A. Lachenbruch and M. R. Mickey. Estimation of Error Rates in Discriminant Analysis. *Technometrics*, 10(1):1–11, 1968.
- [32] M. Lavielle. Using penalized contrasts for the change-point problem. Technical Report 5339, INRIA, 2004.
- [33] M. Lavielle. Using penalized contrasts for the change-point problem. *Signal Proces.*, 85:1501–1510, 2005.
- [34] M. Lavielle and G. Teyssière. Detection of Multiple Change-Points in Multivariate Time Series. *Lithuanian Mathematical Journal*, 46:287–306, 2006.
- [35] E. Lebarbier. Detecting multiple change-points in the mean of a Gaussian process by model selection. *Signal Proc.*, 85:717–736, 2005.
- [36] K.-C. Li. Asymptotic Optimality for  $C_p$ ,  $C_L$ , Cross-Validation and Generalized Cross-Validation: Discrete Index Set. *The Annals of Statistics*, 15(3):958–975, 1987.
- [37] C. L. Mallows. Some comments on  $C_p$ . *Technometrics*, 15:661–675, 1973.
- [38] B. Q. Mia and L. C. Zhao. On detection of change points when the number is unknown. *Chinese J. Appl. Probab. Statist.*, 9(2):138–145, 1993.
- [39] D. Picard. Testing and estimating change points in time series. *J. Appl. Probab.*, 17:841–867, 1985.
- [40] F. Picard. *Process segmentation/clustering Application to the analysis of array CGH data*. PhD thesis, Université Paris-Sud 11, 2005.
- [41] F. Picard, S. Robin, M. Lavielle, C. Vaisse, and J-J. Daudin. A statistical approach for array CGH data analysis. *BMC Bioinformatics*, 27(6):electronic access, 2005.
- [42] Franck Picard, Stéphane Robin, Émilie Lebarbier, and Jean-Jacques Daudin. A segmentation/clustering model for the analysis of array cgh data. *Biometrics*, 2007. To appear. doi:10.1111/j.1541-0420.2006.00729.x.

- [43] M. H. Quenouille. Approximate tests of correlation in time series. *J. Royal Statist. Soc. Series B*, 11:68–84, 1949.
- [44] J. Rissanen. Universal Prior for Integers and Estimation by Minimum Description Length. *The Annals of Statistics*, 11(2):416–431, 1983.
- [45] G. Schwarz. Estimating the dimension of a model. *The Annals of Statistics*, 6(2):461–464, 1978.
- [46] J. Shao. An asymptotic theory for linear model selection. *Statistica Sinica*, 7:221–264, 1997.
- [47] R. Shibata. An optimal selection of regression variables. *Biometrika*, 68:45–54, 1981.
- [48] C.J. Stone. An asymptotically optimal window selection rule for kernel density estimates. *The Annals of Statistics*, 12(4):1285–1297, 1984.
- [49] M. Stone. Cross-validatory choice and assessment of statistical predictions. *J. Roy. Statist. Soc. Ser. B*, 36:111–147, 1974. With discussion by G. A. Barnard, A. C. Atkinson, L. K. Chan, A. P. Dawid, F. Downton, J. Dickey, A. G. Baker, O. Barndorff-Nielsen, D. R. Cox, S. Giesser, D. Hinkley, R. R. Hocking, and A. S. Young, and with a reply by the authors.
- [50] M. Stone. An Asymptotic Equivalence of Choice of Model by Cross-validation and Akaike’s Criterion. *JRSS B*, 39(1):44–47, 1977.
- [51] R. Tibshirani and K. Knight. The Covariance Inflation Criterion for Adaptive Model Selection. *JRSS B*, 61(3):529–546, 1999.
- [52] Y. Yang. Regression with multiple candidate model: selection or mixing? *Statist. Sinica*, 13:783–809, 2003.
- [53] Y. Yang. Comparing Learning Methods for Classification. *Statistica Sinica*, 16:635–657, 2006.
- [54] Y. Yao. Estimating the number of change-points via Schwarz criterion. *Statist. Probab. Lett.*, 6:181–189, 1988.