Learning Sparse Penalties for Change-Point Detection using Max Margin Interval Regression

Guillem Rigaill* RIGAILL@EVRY.INRA.FR
Unité de Recherche en Génomique Végétale INRA-CNRS-Université d'Evry Val d'Essonne, Evry, France

Toby Dylan Hocking* TOBY.HOCKING@INRIA.FR
Francis Bach FRANCIS.BACH@INRIA.FR
INRIA – Sierra project-team, Département d'Informatique de l'École Normale Supérieure, Paris, France

Jean-Philippe Vert JEAN-PHILIPPE.VERT@MINES.ORG
Mines ParisTech – CBIO, INSERM U900, Institut Curie, Paris, France

Abstract

In segmentation models, the number of change-points is typically chosen using a penalized cost function. In this work, we propose to learn the penalty and its constants in databases of signals with weak change-point annotations. We propose a convex relaxation for the resulting interval regression problem, and solve it using accelerated proximal gradient methods. We show that this method achieves state-of-the-art change-point detection in a database of annotated DNA copy number profiles from neuroblastoma tumors.

1. Introduction

Segmentation and change-point detection problems arise in many scientific domains such as climatology, econometrics, molecular biology, machine learning or signal processing (Bai & Perron, 2003; Braun et al., 2000; Venkatraman & Olshen, 2007; Vert & Bleakley, 2010; Harchaoui & Levy-Leduc, 2008; Tibshirani & Wang, 2008; Gillet et al., 2007). When the data to segment $y \in \mathbb{R}^d$ are a 1-dimensional series of length d and the errors are normally distibuted, maximum likelihood inference is constrained least squares:

$$\begin{aligned} \hat{y}^k = \underset{\mu \in \mathbb{R}^d}{\arg\min} \ ||y-\mu||_2^2 \quad , \\ \text{subject to} \ \sum_{j=1}^{d-1} 1_{\mu_j \neq \mu_{j+1}} \leq k-1. \end{aligned} \tag{1}$$

For a given number of segments k there are several efficient algorithms that recover the optimal segmentation (Auger & Lawrence, 1989; Bai & Perron, 2003; Jackson et al., 2005; Rigaill, 2010; Killick et al., 2011).

However in most applications the number of segments k is not known in advance and needs to be determined from the data. To solve this critical problem, many penalty functions specifically adapted to change-point models have been proposed. For example, there are many different variants of the BIC (Yao, 1988; Lee, 1995; Zhang & Siegmund, 2007), the model selection theory of Birgé and Massart suggests another penalty (Birgé & Massart, 2007; Lavielle, 2005; Lebarbier, 2005), and Baraud et al. (2009) proposed another criterion. The formula of these penalties depend on assumptions such as Gaussianity or independence. These assumptions are often violated in real data, which can lead to selection of a suboptimal model.

Hocking et al. (2012) proposed a different approach for selecting the number of segments k: first create a database of change-point annotations, then select a scalar penalty constant that minimizes the change-point detection error. In this article we generalize that approach by learning a multivariate penalty function that agrees with the change-point annotations.

Our method proceeds in essentially two steps. In the first step, we pre-process the signals to obtain the best segmentation (1) for several model sizes k, then calculate a function E_i that maps the penalty value to the annotation error. In the second step, we learn a function f that predicts penalty values which minimize the annotation error E_i. Since explicitly learn-

Proceedings of the 30th International Conference on Machine Learning, Atlanta, Georgia, USA, 2013. JMLR: W&CP volume 28.

*Contributed equally.

ing this function involves an intractable optimization, we instead treat it as an interval regression problem and propose a convex relaxation. We solve the resulting non-smooth optimization problem using accelated proximal gradient methods, which permit efficient inference of the support and constants in the penalty function. For a new un-annotated signal, the learned penalty function can predict the optimal number of segments k.

2. The penalty learning problem

Assume we have a set of n annotated training signals. For every training signal $i \in \{1, \dots, n\}$, let $y_i \in \mathbb{R}^{d_i}$ be the noisy signal sampled at positions $p_i \in \mathbb{N}^{d_i}$, a vector of positive integers sorted such that $p_{i1} < \cdots < p_{i,d_i}$. As shown in the left panel of Figure 1, the positions p_i may not be evenly spaced. Since the points sampled per signal d_i is variable, the $p_1 \in \mathbb{N}^{d_1}, \dots, p_n \in \mathbb{N}^{d_n}$ vectors are not the same size.

We use pruned dynamic programming (DP) to calculate a segmented signal $\hat{y}_i^k \in \mathbb{R}^{d_i}$ as the solution of (1) for each model size $k \in \{1, \dots, k_{\max}\}$ (Rigaill, 2010). The indices where $\hat{y}_i^k$ changes are

$$J_i^k = \left\{ j \in \{1, \dots, d_i - 1\} \mid \hat{y}_{ij}^k \neq \hat{y}_{i,j+1}^k \right\}, \quad (2)$$

and as shown with vertical black lines in the left panel of Figure 1, change-point positions are estimated using

$$\hat{P}_i^k = \{ \lfloor (p_{ij} + p_{i,j+1})/2 \rfloor \mid j \in J_i^k \}. \quad (3)$$

2.1. Change-point annotations define a non-convex error function

As shown in Figure 1, for every signal i, we have a set of regions R_i and corresponding annotations A_i. Every annotation $a \in A_i$ is a set that specifies the expected number of changes in the corresponding region $r \in R_i$. The annotation error $e_i : \{1, \dots, k_{\max}\} \to \mathbb{R}^+$ compares the estimated number of changes in each region $|\hat{P}_i^k \cap r|$ to the annotated number of changes a using the zero-one loss:

$$e_i(k) = \sum_{(r,a) \in (R_i, A_i)} 1_{|\hat{P}_i^k \cap r| \notin a}. \quad (4)$$

For every signal i, we define the optimal number of segments as

$$k_i^*(g) = \operatorname*{arg\,min}_{k \in \{1, \dots, k_{\max}\}} ||y_i - \hat{y}_i^k||_2^2 + g(\hat{y}_i^k, x_i), \quad (5)$$

where the penalty g is a function of the segmentaton $\hat{y}_i^k$ and some features $x_i \in \mathbb{R}^m$ such as signal size or estimated variance. The problem we tackle in this article is to use the n annotated signals to learn the best penalty g for change-point detection, defined as:

$$\min_g \sum_{i=1}^n e_i \left[k_i^*(g) \right]. \quad (6)$$

We will consider penalty functions g that factorize as a model complexity term $h(\hat{y}_i^k, x_i)$ that is given, and a smoothing term $\lambda = \exp f(x_i)$ that we want to learn: $g(\hat{y}_i^k, x_i) = h(\hat{y}_i^k, x_i) \exp f(x_i)$. The exponential forces the smoothing term to be positive. Many existing penalties can be written in this form (see Table 1), and it allows efficient learning by first calculating an exact representation of

$$\hat{k}_i(\lambda) = \operatorname*{arg\,min}_{k \in \{1, \dots, k_{\max}\}} ||y_i - \hat{y}_i^k||_2^2 + \lambda h(\hat{y}_i^k, x_i). \quad (7)$$

The function $\hat{k}_i : \mathbb{R}^+ \to \{1, \dots, k_{\max}\}$ is used to select the number of segments for signal i. In the right panel of Figure 1, we show one function $\hat{k}_i$, and its corresponding annotation error $E_i : \mathbb{R}^+ \to \mathbb{R}^+$, defined as

$$E_i(\lambda) = e_i \left[\hat{k}_i(\lambda) \right]. \quad (8)$$

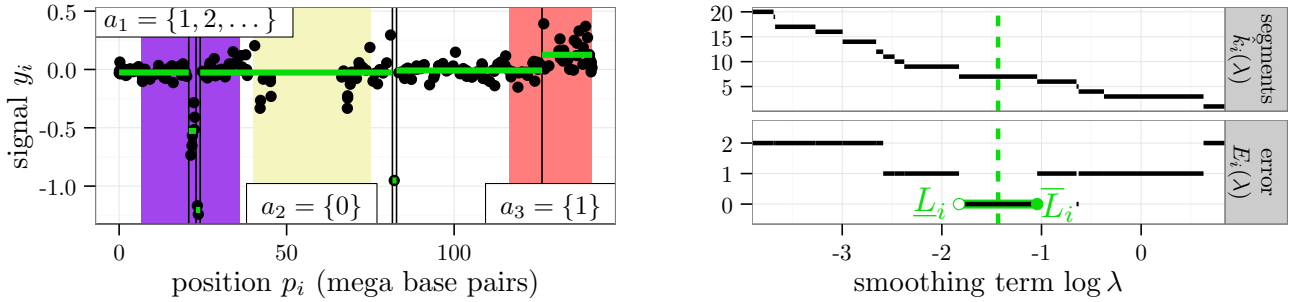

Figure 1. **Left**: black points show one signal (p_i, y_i) with its annotated regions R_i. Counts of allowed change-points $a \in A_i$ are shown as sets. A consistent model $\hat{y}_i^7$ is drawn in green and its change-points $\hat{P}_i^7$ are shown as vertical black lines. **Right**: the optimal number of segments $\hat{k}_i$ and annotation error E_i. The limits $\underline{L}_i < \overline{L}_i$ of the target interval are drawn in green, with a vertical dashed line to indicate the complexity of the model $\hat{y}_i^7$ plotted on the left.

Note that $\hat{k}_i$ and E_i are non-convex, piecewise constant functions that can be efficiently calculated prior to learning, using Algorithm 1 in Section 4.1.

By definition, for a given model complexity term h, we have the following relation between the annotation error functions:

$$e_i\left[k_i^*\left(g\right)\right] = e_i\left[\hat{k}_i\big(\exp f(x_i)\big)\right] \quad (9)$$
$$= E_i\left[\exp f(x_i)\right]. \quad (10)$$

Rewriting learning problem (6) using E_i, we obtain

$$\min_f \sum_{i=1}^n E_i\left[\exp f(x_i)\right]. \quad (11)$$

2.2. Affine smoothing functions

We need to specify what kind of smoothing function $f : \mathbb{R}^m \to \mathbb{R}$ we will learn. Although non-parametric models such as k-nearest neighbors could be used, there are several interesting penalties defined by supposing that f is affine, $f(x_i) = w'x_i + \beta$. This results in the following model selection criterion:

$$\hat{k}_i\left[e^{f(x_i)}\right] = \arg\min_k ||y_i - \hat{y}_i^k||_2^2 + e^{\beta + w'x_i} h(\hat{y}_i^k, x_i). \quad (12)$$

In Table 1, we compare several model selection criteria that are special cases of (12).

For example, the well-known BIC due to Schwarz (1978) uses $h(\hat{y}_i^k, x_i) = k \log d_i$ as a complexity term, and the smoothing term $\exp f(x_i) = 1$ contains no parameters to learn. Instead, we can use a difference-based estimator of signal variance σ_i (Hall et al., 1990), and take $x_i = \log \sigma_i$. Then we can learn α, w_1 in the corresponding smoothing term $\exp f(x_i) = \alpha\sigma_i^{w_1}$.

A similar criterion was suggested by Lebarbier (2005), and its complexity is $k \log(2 \log(d_i/k) + 5)$. Again we can learn a smoothing term that depends on the noise σ_i, which corresponds to choosing $x_i = \log \sigma_i$.

As another example, Zhang & Siegmund (2007) proposed a modified BIC (mBIC) which has a model complexity term of the form $\sum_r \log(n_r) + (2k-1)\log(d_i)$, where n_r is the length of a segment. A default smoothing term with no parameters to learn is implemented in the `uniseg` function of R package `cghseg` (Picard et al., 2012). Our approach can also be used with model complexity terms of this form. If we take two features $x_i = \begin{bmatrix} \log \sigma_i & \log d_i \end{bmatrix}$, that implies a smoothing term $\lambda = \alpha\sigma_i^{w_1} d_i^{w_2}$.

However, inferring the optimal weights w and intercept β in all these models involves an intractable optimization. Since the annotation error E_i is piecewise constant, the minimization in problem (11) can only be accomplished via exhaustive search. For one or two features this may be feasible using grid search. But for multivariate models, grid search is very inefficient. So instead of minimizing the annotation error E_i directly, we propose a convex relaxation in the next section that yields an efficient interval regression algorithm for finding the optimal model parameters.

3. A convex relaxation of the annotation error

In this section, we develop a surrogate loss l_i that is a convex relaxation of the annotation error E_i. In particular, we propose to make learning problem (11) tractable using these two modifications:

- Instead of minimizing $E_i(\lambda)$ directly, we define a target interval of λ values, yielding an interval regression problem.
- We replace the non-convex annotation error E_i with a margin-based convex surrogate loss l_i.

3.1. The interval regression problem

Recall that the goal is to learn $\exp f(x_i) = \lambda$ to minimize the annotation error $E_i(\lambda)$, which is a piecewise constant function that can be calculated exactly using Algorithm 1. So a perfect function f would verify $\exp f(x_i) = \arg\min_\lambda E_i(\lambda)$ for all signals i. Thus we define the target interval $(\underline{L}_i, \overline{L}_i)$ as the largest interval such that $\lambda^* = \arg\min_\lambda E_i(\lambda)$ for all $\log \lambda^* \in (\underline{L}_i, \overline{L}_i)$. The target interval may be closed as shown in the right

Penalty	Complexity term $h(\hat{y}_i^k, x_i)$	Smoothing term $\lambda = \exp f(x_i)$	Learned parameters	Features x_i
BIC	$k \log d_i$	$\alpha\sigma_i^{w_1}$	$\alpha \in \mathbb{R}^+, w_1 \in \mathbb{R}$	$\log \sigma_i$
Lebarbier	$k(c_1 \log(d_i/k) + c_2)$	$\alpha\sigma_i^{w_1}$	$\alpha \in \mathbb{R}^+, w_1 \in \mathbb{R}$	$\log \sigma_i$
mBIC	$\sum_r \log(n_r) + (2k-1)\log(d_i)$	$\alpha\sigma_i^{w_1} d_i^{w_2}$	$\alpha \in \mathbb{R}^+, w_1 \in \mathbb{R}, w_2 \in \mathbb{R}$	$\log \sigma_i, \log d_i$
Lavielle	k	$\alpha\sigma_i^{w_1} d_i^{w_2}$	$\alpha \in \mathbb{R}^+, w_1 \in \mathbb{R}, w_2 \in \mathbb{R}$	$\log \sigma_i, \log d_i$
General	$h(\hat{y}_i^k, x_i)$	$\exp\{x_i' w + \beta\}$	$\beta = \log \alpha \in \mathbb{R}, w \in \mathbb{R}^m$	$x_i \in \mathbb{R}^m$

Table 1. Some penalties that we can learn using affine smoothing functions f.

panel of Figure 1, or open ($\underline{L}_i = -\infty$ or $\overline{L}_i = \infty$), as shown in the top 2 functions in Figure 3.

In summary, for every signal i, $f(x_i) \in (\underline{L}_i, \overline{L}_i)$ implies that $\exp f(x_i) = \arg\min_\lambda E_i(\lambda)$. This is an interval regression problem since predicting any value in the target interval has the same minimal error.

There is an equivalent geometric interpretation of the learning problem in terms of the target interval. In the middle panel of Figure 3 we plot the target intervals $(\underline{L}_i, \overline{L}_i)$ as a function of one feature, a variance estimate $x_i = \log \sigma_i$. Geometrically, the learning problem corresponds to finding a line f that intersects each of the target intervals.

Another interpretation is shown in the bottom panel of Figure 3, where we plot just the limits $\underline{L}_i, \overline{L}_i$ of the target interval. The learning problem corresponds to finding a line f that separates the two classes of points.

3.2. Maximum margin regression line for separable data

If there are few data as in Figure 2, then it may be possible to find a regression function $f(x_i) = w'x_i + \beta$ such that $\underline{L}_i < f(x_i) < \overline{L}_i$ for all signals i. In this case the data are separable, and in fact there are infinitely many functions f that satisfy these criteria. However, for learning it is best to use the max margin separator:

$$\begin{aligned} \underset{\beta\in\mathbb{R}, w\in\mathbb{R}^m, \mu\in\mathbb{R}^+}{\text{maximize}} \quad & \mu \\ \text{subject to} \quad & \forall i,\ \text{if } \underline{L}_i > -\infty, \quad w'x_i + \beta - \underline{L}_i \geq \mu \\ & \forall i,\ \text{if } \overline{L}_i < \infty, \quad \overline{L}_i - w'x_i - \beta \geq \mu. \end{aligned} \tag{13}$$

Since the objective and the constraints are linear, this is a linear program (LP), so any LP solver can be used.

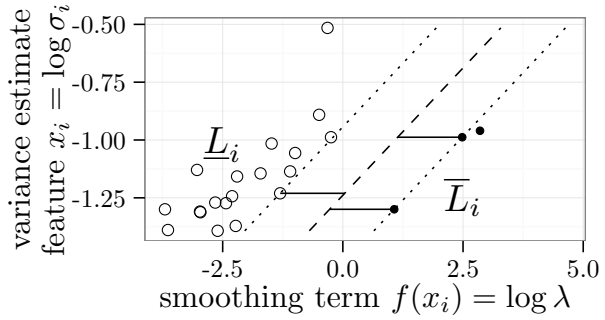

Figure 2. The maximum margin interval regression line $f(x_i)$ is found by solving problem (13), and is drawn as a dashed line. The limits $\underline{L}_i, \overline{L}_i$ of target intervals are drawn using points for a small data set that is linearly separable using the variance estimate feature $x_i = \log \sigma_i$. The horizontal margin μ is drawn for the 3 border points.

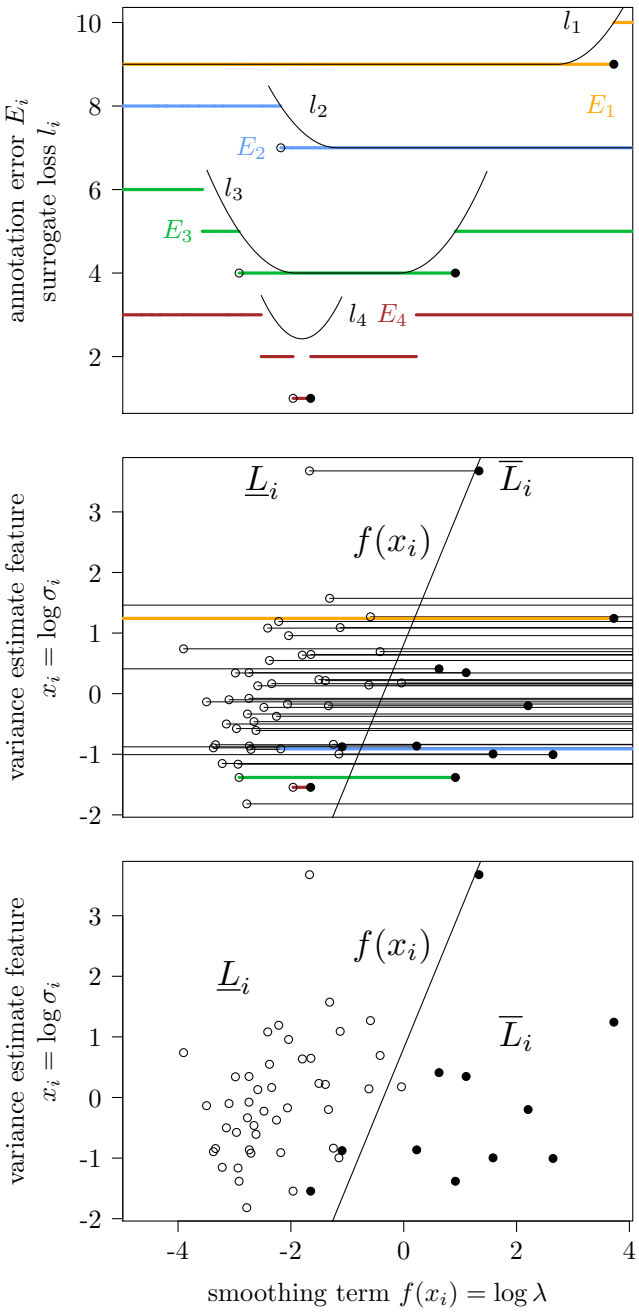

Figure 3. **Top**: the surrogate loss l_i from Equation (14) in black along with the annotation error E_i in color, for 4 signals i. The quadratic tails of the convex surrogate loss are not shown. **Middle**: for several more signals i, we show only the target interval $(\underline{L}_i, \overline{L}_i)$, plotted using a variance estimate feature $x_i = \log \sigma_i$ on the vertical axis. The regression line is found by minimizing the average surrogate loss over all signals (16). **Bottom**: only the minimum $\underline{L}_i$ and maximum $\overline{L}_i$ of each target interval is shown.

The regression function found by solving problem (13) for a small separable data set with 1 feature is shown in Figure 2. It is important to note that the geometric interpretation of the margin is not the same as the usual Support Vector Machine for binary classification. In fact, the margin is the distance along the $\log \lambda$ axis between the regression line and the closest limits $\underline{L}_i, \overline{L}_i$.

However, any sizable real data set will not be separable. So in the next section, we develop a surrogate loss for interval regression on non-separable data sets.

3.3. Surrogate loss for non-separable data

We relax the annotation error E_i in the $L = \log \lambda$ space, and consider the class of surrogate loss functions $l_i : \mathbb{R} \to \mathbb{R}^+$ defined by

$$l_i(L) = \varphi\left(\frac{L - \underline{L}_i}{\delta}\right) + \varphi\left(\frac{\overline{L}_i - L}{\delta}\right), \qquad (14)$$

where the binary classification surrogate loss function $\varphi : \mathbb{R} \to \mathbb{R}^+$ is a convex upper bound of the zero-one loss. The parameter $\delta > 0$ controls the size of the margin, and we used $\delta = 1$ since that worked well in the data we analyzed. Using the hinge loss for φ results in a surrogate loss similar to the ϵ-insensitive loss used for Support Vector Regression (Vapnik et al., 1997). Some other choices for φ include log and Huber losses, but we used the squared hinge loss since it exhibited the best empirical performance:

$$\varphi(L) = \begin{cases} (L-1)^2 & \text{if } L \leq 1 \\ 0 & \text{if } L \geq 1. \end{cases} \qquad (15)$$

Note that l_i is convex since it is the sum of two convex functions. This convex relaxation can clearly be seen in the top panel of Figure 3, where we plot the surrogate loss l_i along with the annotation error E_i for several signals i. Let the average surrogate loss be

$$\mathcal{L}(\beta, w) = \frac{1}{n}\sum_{i=1}^{n} l_i(w' x_i + \beta). \qquad (16)$$

For each signal i we have features $x_i \in \mathbb{R}^m$. When m is large we can encourage a sparse weight vector w by using an ℓ_1 penalty, which yields the optimization problem

$$\underset{\beta\in\mathbb{R}, w\in\mathbb{R}^m}{\text{minimize}}\ \gamma||w||_1 + \mathcal{L}(\beta, w), \qquad (17)$$

where $\gamma \in \mathbb{R}^+$ is a fixed value that controls the degree of regularization. Note that the ℓ_1 norm encourages some entries of w to be exactly zero, which has the effect of selecting which features are used in the penalty function $f(x_i) = w' x_i + \beta$.

4. Algorithms

Recall that using pruned DP we obtain $\hat{y}_i^k$ for $k \in \{1, \ldots, k_{\max}\}$, and we use (4) to calculate the annotation error $e_i(k)$. For every signal i we then need to recover the functions $\hat{k}_i$ and E_i so we can calculate the target intervals $(\underline{L}_i, \overline{L}_i)$ and the surrogate loss l_i. First we discuss how to calculate the exact functions $\hat{k}_i, E_i$, then we discuss surrogate loss optimization.

4.1. Exact annotation error as a function of λ

For a given signal i, number of segments k and model complexity h, $\text{crit}_i^k(\lambda) = ||y_i - \hat{y}_i^k||_2^2 + \lambda h(\hat{y}_i^k, x_i)$ is an affine function of λ. Thus $\hat{k}_i(\lambda) = \arg\min_k \text{crit}_i^k(\lambda)$ is the minimum of a finite set of affine functions, which we calculate exactly using path-following Algorithm 1. The result is a list of λ values for which there is a change in the optimal number of segments $\hat{k}_i$.

We use the following Lemma to exclude some model sizes k' that will never be selected using $\hat{k}_i$.

Lemma 1. *If $k' < k$ and $h(\hat{y}_i^k, x_i) \leq h(\hat{y}_i^{k'}, x_i)$ then for all $\lambda \geq 0$, we have $\hat{k}_i(\lambda) \neq k'$.*

Proof. The squared error $||y_i - \hat{y}_i^k||_2^2$ is a decreasing function of k, so $\text{crit}_i^k(\lambda) < \text{crit}_i^{k'}(\lambda)$ for all $\lambda \geq 0$. Thus $k' \neq \arg\min_k \text{crit}_i^k(\lambda) = \hat{k}_i(\lambda)$. □

The complexity of Algorithm 1 is $O(k_{\max}^2)$. First, the initial set of *plausibleK* is defined using Lemma 1. Each iteration of the while loop finds the smallest λ for which $k \in$ *plausibleK* is preferred over the current k_c. The result is an exact representation of the piecewise constant function $\hat{k}_i$ via its breakpoints k, λ. We then

Algorithm 1 Exact recovery of $\hat{k}_i$

Input: $||y_i - \hat{y}_i^k||_2^2, h(\hat{y}_i^k, x_i), \forall k \in \{1, \ldots, k_{\max}\}$.
$k_c \leftarrow \max\{\textit{plausibleK}\}$
$\textit{plausibleK} \leftarrow \textit{plausibleK} \setminus k_c$
while *plausibleK* $\neq \emptyset$ **do**
  $\text{next}\lambda \leftarrow +\infty, \quad \text{nextK} \leftarrow 0$
  for $k \in$ *plausibleK* **do**
    $$\text{hit\_time} \leftarrow \frac{||y_i - \hat{y}_i^{k_c}||_2^2 - ||y_i - \hat{y}_i^k||_2^2}{h\left(\hat{y}_i^k, x_i\right) - h\left(\hat{y}_i^{k_c}, x_i\right)}$$
    if $\text{next}\lambda > \text{hit\_time}$ **then**
      $\text{next}\lambda \leftarrow \text{hit\_time}, \quad \text{nextK} \leftarrow k$
    end if
  end for
  $k_c \leftarrow \text{nextK}, \quad$ **Save** $k_c, \text{next}\lambda$
  $\textit{plausibleK} \leftarrow \textit{plausibleK} \setminus \{k \mid k \geq k_c\}$
end while
Output: $\hat{k}_i$ represented by breakpoints $k_c, \text{next}\lambda$.

use definition (8) to recover the annotation error E_i.

4.2. Surrogate loss optimization using FISTA

The learning problem (17) is to find the affine function $f(x_i) = w'x_i + \beta$ that minimizes the sum of a non-smooth convex penalty and a smooth convex surrogate loss. So we can solve it using proximal gradient methods such as FISTA, a Fast Iterative Shrinkage-Thresholding Algorithm (Beck & Teboulle, 2009). We need the partial derivatives of the surrogate loss:

$$\frac{\partial}{\partial \beta} l_i\left[f(x_i)\right] = \varphi'\left[f(x_i) - \underline{L}_i\right] - \varphi'\left[\overline{L}_i - f(x_i)\right] \quad (18)$$

$$\frac{\partial}{\partial w_j} l_i\left[f(x_i)\right] = x_{ij}\left(\varphi'\left[f(x_i) - \underline{L}_i\right] - \varphi'\left[\overline{L}_i - f(x_i)\right]\right), \quad (19)$$

where the derivative of the squared hinge loss is

$$\varphi'(L) = \begin{cases} 2(L-1) & \text{if } L \leq 1 \\ 0 & \text{if } L \geq 1. \end{cases} \quad (20)$$

The proximal operator $p_\eta : \mathbb{R}^{m+1} \to \mathbb{R}^{m+1}$ is

$$p_\eta(\beta, w) = \begin{bmatrix} \beta - \frac{1}{\eta}\frac{\partial}{\partial \beta}\mathcal{L}(\beta, w) \\ s_{\gamma/\eta}\left(w_1 - \frac{1}{\eta}\frac{\partial}{\partial w_1}\mathcal{L}(\beta, w)\right) \\ \vdots \end{bmatrix} \quad (21)$$

where s_λ is the soft-thresholding function and η is a Lipschitz constant of the smooth loss (Beck & Teboulle, 2009). We use a constant $\eta = m + \sqrt{m}$ which is heuristic but worked well on the data we analyzed. Importantly, the assumptions of FISTA are satisfied, since the squared hinge loss is indeed Lipschitz continuous (Flamary et al., 2012).

After each application of the proximal operator, we check for approximate subdifferential optimality:

$$\left|\frac{\partial}{\partial \beta}\mathcal{L}(\beta, w)\right| \leq \epsilon, \quad (22)$$

and for every feature $j \in \{1, \ldots, m\}$,

$$\begin{cases} \left|\frac{\partial}{\partial w_j}\mathcal{L}(\beta, w) - \gamma\right| \leq \epsilon & \text{if } w_j < 0 \\ \left(\left|\frac{\partial}{\partial w_j}\mathcal{L}(\beta, w)\right| - \gamma\right)_+ \leq \epsilon & \text{if } w_j = 0 \\ \left|\frac{\partial}{\partial w_j}\mathcal{L}(\beta, w) + \gamma\right| \leq \epsilon & \text{if } w_j > 0 \end{cases} \quad (23)$$

for some positive constant $\epsilon > 0$ that controls how far we are from an optimal solution. For learning it is not necessary to take a very small ϵ (Bottou & Bousquet, 2008), and we found that $\epsilon = 10^{-3}$ is sufficient.

5. Results and discussion

We analyzed 3 data sets of annotated neuroblastoma DNA copy number profiles (Table 2). These data come from a set of 575 chromosomal copy number profiles of tumors taken from children when they were diagnosed. In these data, accurate change-point detection is crucial in order to precisely characterize the genetics of these tumors.

We defined model complexity as the number of segments $h(\hat{y}_i^k, x_i) = k$, which corresponds to a Lavielle penalty (Table 1). We used Algorithm 1 to calculate target intervals for 3 annotation data sets based on the neuroblastoma data, and for our annotation of some simulated data. In Figure 4, the scatterplots of target intervals show a clear dependence on the estimated noise σ_i, which is not modeled using the state-of-the-art cghseg.k model (Hocking et al., 2012).

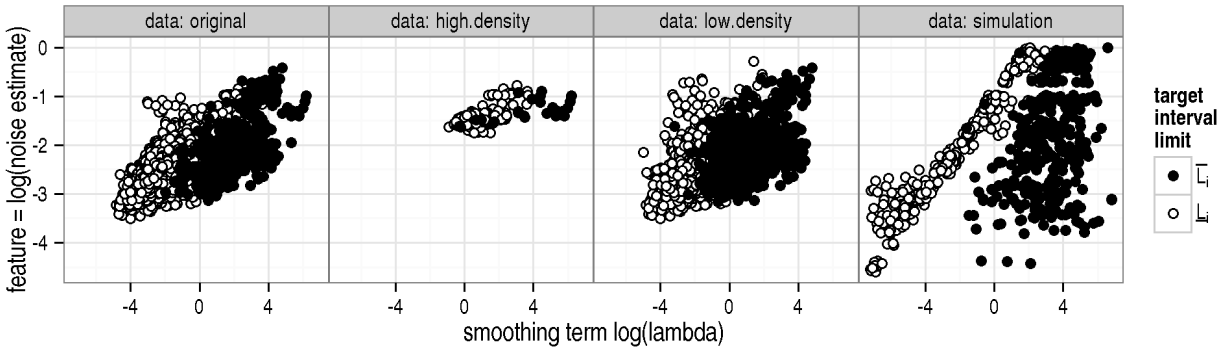

Figure 4. Algorithm 1 was used to calculate the limits $\underline{L}_i, \overline{L}_i$ of the target interval, which are plotted against a variance estimate feature $x_i = \log \sigma_i$ for all signals i in four annotated data sets. The original data are taken from R package neuroblastoma, and the high and low density data sets are two other annotations of the neuroblastoma data. The simulation data are a simulated set of signals with Gaussian noise, annotated by expert visual inspection.

5.1. Annotation protocols

Our penalty learning algorithms rely on the quality of the annotations, which come from either prior knowledge or expert visual inspection. We consider four annotation data sets (Table 2), constructed using two protocols for expert visual annotation:

- **(Systematic)** For the "original" annotations, a set of regions was defined, then the expert systematically examined and annotated these regions on each signal i.
- **(Any)** For the other annotation data sets, the expert was asked to browse all signals and draw rectangles around any regions where she was sure of the annotation.

We include these details because we conjecture that the ability of the learning algorithm may be limited by the annotation protocol.

5.2. Accuracy of annotations in simulations

To assess the quality of the visual annotations, we simulated Gaussian signals with different segment length, noise and change size, and annotated them using the **Any** protocol. By comparing the latent signal in the simulation with the manual annotations, we observed the following results:

- Out of 697 regions annotated to have 1 change-point, 24 contained 2–4 changes.
- Out of 147 regions annotated to have no change-points, 27 contained 1–2 changes.
- These false-negative changes had a low signal to noise ratio, so they are not detected in any case using maximum likelihood segmentation.

One may fear that the **Any** protocol would result in a database of "easy" annotated changes with a high signal-to-noise ratio. However, we did not observe that when comparing the distribution of annotated changes to the distribution of all changes (t-test, KS-test, and Wilcoxon test). So we concluded that visual annotations are indeed useful for recovering significant, detectable change-points.

5.3. Learned penalty functions

We learned penalty functions on each of the four data sets (Table 2) using four models (Table 3). Recall the Lavielle model from Table 1:

$$f(x_i) = \beta + w_1 \log \sigma_i + w_2 \log d_i. \tag{24}$$

We compared 3 un-regularized versions of this model, and one ℓ_1-regularized model with 117 features.

The **cghseg.k** model uses 0 features, takes $w_1 = 0$ and $w_2 = 1$, then learns β by solving (11) with grid search (Hocking et al., 2012).

The **log.d** model uses 1 feature $\log d_i$, takes $w_1 = 0$, and learns w_2 and β by minimizing the un-regularized surrogate loss (16).

The **log.s.log.d** model uses 2 features $\log d_i, \log \sigma_i$ and learns w_1, w_2, β by minimizing the un-regularized surrogate loss. We report the coefficients learned in this model in Table 2, and it is interesting to note that the coefficients are clearly not the same across data sets.

In Table 2 the optimal penalty for the original data contains a $d_i^{0.96}$ term. This is in agreement with the observation that the cghseg.k model has a d_i^1 term and works well in these data (Hocking et al., 2012).

In Table 2 it is clear that $w_1 \neq 2$, which means the penalties do not contain the σ_i^2 term that is suggested by model selection theory. This is evidence that theoretical arguments are not sufficient for good change-point detection in real data, as measured by visual annotations.

L1-reg constructs a feature vector $x_i \in \mathbb{R}^{117}$ consisting of features such as variance estimates, signal size measurements (d_i, $\log d_i$, ...), model RSS and MSE, and indicator variables for each chromosome. We use

	n	anns	pro.	noise σ_i	points d_i	noise w_1	points w_2	intercept β
original	3418	3418	Sys.	0.03 – 0.66	66 – 5937	1.01 ± 0.03	0.96 ± 0.02	-2.66 ± 0.10
high.density	204	210	Any	0.17 – 0.46	1948 – 5937	3.16 ± 0.38	0.08 ± 0.26	6.54 ± 2.38
low.density	3542	4171	Any	0.03 – 0.76	25 – 657	1.30 ± 0.02	0.93 ± 0.02	-2.00 ± 0.13
simulation	377	844	Any	0.01 – 1.00	1000 – 2000	1.76 ± 0.07	1.20 ± 0.16	-5.33 ± 1.20

Table 2. Several features of the four annotation data sets. We show the number of signals n, the number of annotations (anns), and the annotation protocol as explained in Section 5.1. The ranges of noise estimates σ_i and signal sizes d_i are shown along with the coefficients of the log.s.log.d model (24), which were estimated by minimizing (16). Signals were split into 10 folds, and we report the mean and standard deviation of coefficients over 10 training sets of size $9n/10$.

V-fold cross-validation to pick the regularization γ. For each training set we first form the standardized features $X \in \mathbb{R}^{n\times m}$ to solve problem (17) with a small $\gamma > 0$. After finding an optimal solution, we increase γ and use a warm restart to find the next optimal solution in the path. We stop after finding a γ for which all coefficients $w_j = 0$. The model that gives minimal annotation error on test fold v is saved as $\hat{\gamma}_v$, and finally we take the mean across folds: $\sum_{v=1}^{V} \hat{\gamma}_v / V$.

5.4. Change-point detection accuracy

We used cross-validation to compare the four models, and the test annotation error is shown in Table 3. First, the standard BIC and mBIC model selection criteria do not use the change-point annotations, so yield error rates much higher than the other models. The only exception is the mBIC in the simulated data set, which is expected since the theoretical conditions of the mBIC are perfectly met in that case.

The log.d model that minimizes the surrogate loss shows comparable performance to cghseg.k, which uses grid search to directly minimize the non-convex annotation error E_i. Both of these methods ignore the noise σ_i, so in general yield sub-optimal change-point detection. The only exception is the high.density data set, in which all learning methods perform about the same, since the noise is relatively uniform (Figure 4 and Table 2).

Table 3 also shows that the 117-feature L1-reg model performs comparably to the 2-feature log.s.log.d model. This suggests that the signal noise σ_i and number of points d_i are sufficient to learn a penalty for optimal change-point detection in these data sets.

However, training the L1-reg model is very time-consuming since an internal cross-validation loop is used to select the degree of regularization γ. So to quickly learn a penalty for data like these, we suggest learning the log.s.log.d model (24) by minimizing the un-regularized surrogate loss (16).

6. Conclusions

We proposed a method to learn an optimal penalty function for change-point detection in databases of annotated signals. Our approach can accomodate most existing model complexity terms (Table 1), and chooses the smoothing term by minimizing a margin-based convex surrogate loss using FISTA. Using our method, one uses an annotation database to tune the parameters of his favorite model selection criterion, yielding penalty terms which are different from those motivated using theoretical arguments (Table 2).

We showed that learning the penalty function using this method results in state-of-the-art change-point detection in several databases of annotated DNA copy number profiles. In particular, standard criteria such as BIC ignore the annotation data so perform much worse than the models we learned (Table 3).

For even better performance, one could use grid search on the support of w found with the L1-reg model to directly optimize the annotation error E_i rather than the surrogate loss l_i. Also, it should be straightforward to apply the kernel trick to learn a penalty which is a non-linear function of the input features. Finally, we may be able to derive efficient algorithms by exploring the duals of the separable and non-separable max-margin interval regression problems.

For future work, we are considering more general penalty functions. For example, Lebarbier (2005) proposed $k(c_1 \log(d_i/k) + c_2)$ and calibrated $c_1 = 2$ and $c_2 = 5$ using a large set of simulated signals. It is reasonable to think that these values of c_1 and c_2 are not optimal for real data and one would like to learn these c_1, c_2 from a database of annotated signals. To learn these more general penalties we are exploring multi-dimensional interval regression.

Acknowledgements: This work was supported by grants DIGITEO-BIOVIZ-2009-25D, SIERRA-ERC-239993, SMAC-ERC-280032, ANR-09-BLAN-0051-04.

model	features m	original	high.density	low.density	simulation
BIC	0	7.99 ± 0.00	19.52 ± 0.00	13.64 ± 0.00	11.97 ± 0.00
mBIC	0	40.99 ± 0.00	70.00 ± 0.00	36.88 ± 0.00	2.25 ± 0.00
cghseg.k	0	2.19 ± 0.82	6.64 ± 3.99	6.49 ± 1.16	11.85 ± 3.52
log.d	1	2.40 ± 1.00	7.59 ± 6.43	6.21 ± 1.01	13.13 ± 4.14
log.s.log.d	2	1.90 ± 0.77	8.12 ± 5.62	4.72 ± 0.54	1.50 ± 1.63
L1-reg	117	1.81 ± 0.58	7.66 ± 5.72	4.70 ± 0.88	1.28 ± 1.47

Table 3. Change-point detection error of models was estimated using 10-fold cross-validation. Means and standard deviations are shown for 4 annotation data sets (columns) and 6 models (rows). The modified Bayesian information criterion (mBIC) and BIC do not use the annotation data, and are defined in Section 2.2. The other models use the annotation data and the indicated number of features to predict model complexity.

References

Auger, I E and Lawrence, C E. Algorithms for the optimal identification of segment neighborhoods. *Bulletin of mathematical biology*, 51(1):39–54, 1989.

Bai, J. and Perron, P. Computation and analysis of multiple structural change models. *J. Appl. Econ.*, 18:1–22, 2003.

Baraud, Y., Giraud, C., and Huet, S. Gaussian model selection with unknown variance. *Ann. Statist.*, 37 (2):630–672, 2009.

Beck, A. and Teboulle, M. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. Imaging Sciences*, 2(1):183–202, 2009.

Birgé, L. and Massart, P. Minimal penalties for gaussian model selection. *Probability Th. and Related Fields*, 138:33–73, 2007.

Bottou, Léon and Bousquet, Olivier. The tradeoffs of large scale learning. In Platt, J.C., Koller, D., Singer, Y., and Roweis, S. (eds.), *NIPS*, pp. 161–168. 2008.

Braun, J. V., Braun, R. K., and Muller, H. G. Multiple changepoint fitting via quasilikelihood, with application to DNA sequence segmentation. *Biometrika*, 87(2):301–314, June 2000.

Flamary, R., Jrad, N., Phlypo, R., Congedo, M., and Rakotomamonjy, A. Mixed-norm regularization for brain decoding. HAL tech. report 00708243, 2012.

Gillet, O., Essid, S., and Richard, G. On the correlation of automatic audio and visual segmentations of music videos. *IEEE Trans. Cir. and Sys. for Video Technol.*, 17(3):347355, March 2007.

Hall, P., Kay, J. W., and Titterinton, D. M. Asymptotically optimal difference-based estimation of variance in nonparametric regression. *Biometrika*, 77 (3):521–528, January 1990.

Harchaoui, Zaid and Levy-Leduc, Céline. Catching change-points with lasso. In Platt, J.C., Koller, D., Singer, Y., and Roweis, S. (eds.), *NIPS*, pp. 617–624. MIT Press, Cambridge, MA, 2008.

Hocking, T.D., Schleiermacher, G, Janoueix-Lerosey, I., Delattre, O., Bach, F., and Vert, J.-P. Learning smoothing models using breakpoint annotations. HAL technical report 00663790, 2012.

Jackson, B., Scargle, J.D., Barnes, D., Arabhi, S., Alt, A., Gioumousis, P., Gwin, E., San, P., Tan, L., and Tsai, Tun Tao. An algorithm for optimal partitioning of data on an interval. *IEEE Signal Processing Letters*, 12(2):105–108, February 2005.

Killick, R., Fearnhead, P., and Eckley, I. A. Optimal detection of changepoints with a linear computational cost. *arXiv:1101.1438*, January 2011.

Lavielle, M. Using penalized contrasts for the change-point problem. *Sig. Proc.*, 85(8):1501–1510, 2005.

Lebarbier, E. Detecting multiple change-points in the mean of gaussian process by model selection. *Signal Processing*, 85:717–736, 2005.

Lee, C.-B. Estimating the number of change points in a sequence of independent normal random variables. *Statist. Proba. Lett.*, 25(3):241–8, 1995.

Picard, Franck, Hoebeke, Mark, Lebarbier, Emilie, Miele, Vincent, Rigaill, Guillem, and Robin, Stephane. *cghseg: Segmentation methods for array CGH analysis*, 2012. R package version 1.0.1.

Rigaill, G. Pruned dynamic programming for optimal multiple change-point detection. arXiv:1004.0887, 2010.

Schwarz, G. Estimating the dimension of a model. *Ann. Statist.*, 6(2):461–4, 1978.

Tibshirani, Robert and Wang, Pei. Spatial smoothing and hot spot detection for CGH data using the fused lasso. *Biostatistics*, 9(1):18–29, January 2008.

Vapnik, V., Golowich, S., and Smola, A. J. Support vector method for function approximation, regression estimation, and signal processing. In Mozer, M. C., Jordan, M. I., and Petsche, T. (eds.), *NIPS*, pp. 281–287, 1997.

Venkatraman, E S and Olshen, Adam B. A faster circular binary segmentation algorithm for the analysis of array CGH data. *Bioinformatics*, 23(6):657–663, March 2007.

Vert, Jean-Philippe and Bleakley, Kevin. Fast detection of multiple change-points shared by many signals using group LARS. In Lafferty, J., Williams, C. K. I., Shawe-Taylor, J., Zemel, R. S., and Cullota, A. (eds.), *NIPS*, pp. 2343–2351, 2010.

Yao, Y.-C. Estimating the number of change-points via Schwarz' criterion. *Statistics & Probability Letters*, 6(3):181–189, February 1988.

Zhang, N. R. and Siegmund, D. O. A modified Bayes information criterion with applications to the analysis of comparative genomic hybridization data. *Biometrics*, 63(1):22–32, 2007.