Estimation and Inference in Changepoint Models

Sean William Jewell

A dissertation
submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy

University of Washington

2020

Reading Committee:

Daniela Witten, Chair

Zaid Harchaoui

Ali Shojaie

Program Authorized to Offer Degree:
Statistics

University of Washington

Abstract

Estimation and Inference in Changepoint Models

Sean William Jewell

Chair of the Supervisory Committee:
Professor Daniela Witten
Departments of Statistics and Biostatistics

This thesis is motivated by statistical challenges that arise in the analysis of calcium imaging data, a new technology in neuroscience that makes it possible to record from huge numbers of neurons at single-neuron resolution. We consider the problem of estimating a neuron's spike times from calcium imaging data. A simple and natural model suggests a non-convex optimization problem for this task. We show that by recasting the non-convex problem as a changepoint detection problem, we can efficiently solve it for the global optimum using a clever dynamic programming strategy.

Furthermore, we introduce a new framework to quantify the uncertainty associated with a set of estimated changepoints in a change-in-mean model. In particular, we propose a new framework to test the null hypothesis that there is no change in mean around an estimated changepoint. This framework can be efficiently carried out in the case of changepoints estimated by binary segmentation and its variants, ℓ_0 segmentation, or the fused lasso, and is valid in finite samples. Our setup allows us to condition on much less information than existing approaches, thereby yielding higher powered tests. These ideas can be generalized to the spike estimation problem.

TABLE OF CONTENTS

LIST OF FIGURES

ACKNOWLEDGMENTS

I would like to thank many people for their support and guidance. None of this work would have been possible without my advisor, Daniela Witten. I would like to thank Daniela for her encouragement, dedication, and mentorship in all aspects of my professional life. I would also like to thank my collaborator, Paul Fearnhead, for his enthusiastic contributions to two projects of this thesis. I would like to thank my collaborator Toby Hocking for his involvement in one project of this thesis. I would like to thank my reading committee members, Zaid Harchaoui and Ali Shojaie, for serving on my committee and providing insightful feedback. I am grateful to the staff, faculty, and students of the Department of Statistics for making Seattle a home away from home. Lastly, I am thankful to my parents and sister for their endless support and to Jenny for always believing in us.

DEDICATION

To my family

Chapter 1

INTRODUCTION

The work in this thesis is inspired by a measurement technology in neuroscience called calcium imaging; discussed in detail in Chapter 2. This technique has opened the door to simultaneous measurement of hundreds or thousands of neurons in behaving animals. To take full advantage of these experimental developments, there is a pressing need for statistical approaches that can map fluorescence traces obtained from calcium imaging to the underlying neuron spike times. Our contributions are two-fold. We introduce new statistical methodology for this task, and facilitate its use in neuroscience experiments conducted at the Allen Institute for Brain Science (AIBS).

1.1 Statistical Methodology Contributions

In Chapter 3, we introduce our approach to estimate spike times based on fluorescence traces obtained from calcium imaging. This approach estimates spike times by solving an ℓ_0 penalized optimization problem. We show that this optimization problem can be efficiently solved for the global optimum using a simple and efficient dynamic programming algorithm. Promising performance is illustrated on small datasets where the ground truth spike times are known. Nonetheless, one drawback of this initial approach is that our naive algorithm cannot be applied to large populations of neurons for computational reasons.

This leads us to develop a fast algorithm for solving the ℓ_0 optimization problem in Chapter 4. Our implementations of this algorithm solve the optimization problem in a fraction of a second on an hour-long fluorescence trace. This amounts to an improvement of three orders of magnitude relative to the original algorithm. Moreover, we show that our new algorithm can easily incorporate additional constraints to prohibit biologically unrealistic

behavior.

Chapter 3 and Chapter 4 provide estimates for the spike times on the basis of a fluorescence trace. Using calibration data, we show that these estimates are "close" to the ground truth spikes. Although such calibration datasets provide a wonderful opportunity to assess model performance, in practice, ground truth spike times are unknown. This leads us to ask, "How certain are we about these spikes?" This is a surprisingly difficult question to answer because we must account for the estimation procedure when determining the null distribution of any test statistic.

In Chapter 5, we introduce a new framework that allows us to quantify the uncertainty associated with an estimated changepoint through a p-value or a confidence interval. Our framework is developed in detail for the change-in-mean problem: we test the null hypothesis that there is no change in mean associated with an estimated changepoint, and show that our approach can be efficiently used for changepoints estimated via binary segmentation and its variants, ℓ_0 segmentation, and the fused lasso.

In Chapter 6, we show that a straightforward extension of this framework can be used to assess the uncertainty in spike estimates from Chapter 3 and Chapter 4. Chapter 6 outlines several interesting avenues for future work.

1.2 Neuroscience Contributions

Our work was motivated by the release of the AIBS Brain Observatory's calcium imaging dataset of "unprecedented size and scope" (Shen, 2016). This initial dataset contained neuronal activity of 18,000 neurons in 25 mice measured over 360 experimental sessions. In each experimental session, calcium imaging was used to measure neuronal activity in the visual cortex of an awake and behaving mouse presented with a visual stimulus. By measuring large populations of neurons in the visual cortex, neuroscientists aim to understand how the brain processes visual stimuli.

The first step in this process is to map the fluorescence traces obtained from calcium imaging to the underlying neuron spike times. Our very fast algorithm from Chapter 4 has

been used to estimate spike times for the entire AIBS Brain Observatory, which as of this writing contains nearly 60,000 neurons. In the platform paper describing the AIBS Brain Observatory, on which I am a co-author, all scientific analyses rely on the output of our algorithm (de Vries et al., 2020). Moreover, the AIBS makes the estimated spikes obtained using our algorithm available to the public through their data platform for all of the calcium imaging data.

Extensions of our testing framework in Chapter 5 to the spike estimation problem, described in Chapter 6, are positioned to have a direct impact on real-world analyses of calcium imaging data. In particular, this framework will allow neuroscientists to propagate the uncertainty in spike estimates to downstream analyses.

Chapter 2

CALCIUM IMAGING PRELIMINARIES

This chapter is based on Jewell and Witten (2018) and Jewell et al. (2019b).

To understand how sensory information is processed by neural circuits, and subsequently how this information is used to implement motor transformations, spontaneous activity, cognitive function, or behavior, neuroscientists are pursuing technologies to measure large populations of neurons in behaving animals (Ahrens et al., 2013; Prevedel et al., 2014). These recordings attempt to capture potential interactions among neurons that are hard to determine in studies of small populations of neurons (Dombeck et al., 2007).

In contrast to traditional approaches, calcium imaging is designed to measure from large areas of the brain at single neuron resolution. The basic idea is simple to describe: When a neuron spikes, calcium floods the cell. In order to quantify intracellular calcium levels, calcium imaging techniques make use of fluorescent calcium indicator molecules that fluoresce when a neuron fires (Dombeck et al., 2007; Ahrens et al., 2013; Prevedel et al., 2014). To capture these events, cameras are surgically attached to the brain of animals and are used to record movies of neuron activity under varying experimental conditions. Remarkably, these videos show groups of pixels—corresponding to the location of neurons—fluorescing as neurons fire.

Determining individual neuron firing times on the basis of these movies is typically a two step process. In the first step, pixels from the movie are segmented to identify neurons. For each identified neuron i, a fluorescence trace is calculated based on the fluorescence intensity of the pixels assigned to neuron i in step one. This is a first-order approximation of the neuron's activity level over time. The second step is inferring the spike times on the basis of a fluorescence trace; this has been the focus of substantial investigation (Grewe et al.,

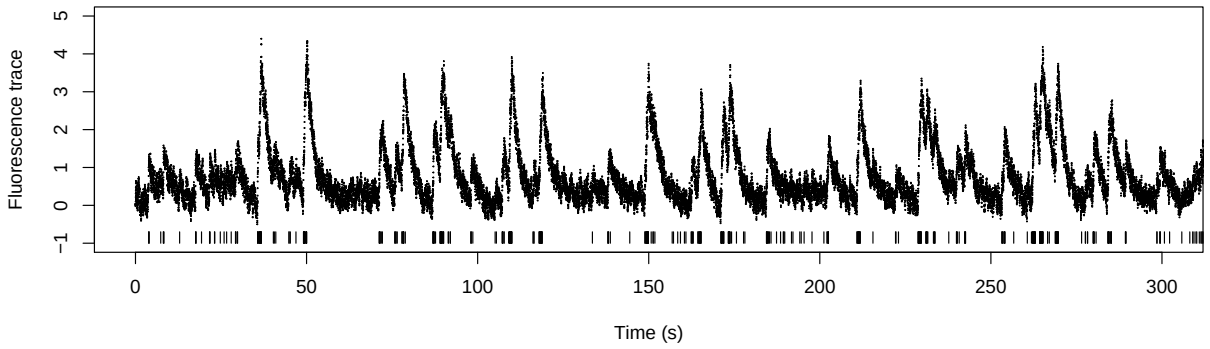


Figure 2.1: Fluorescence trace from cell five of `spikefinder` dataset one. The fluorescence trace obtained using calcium imaging is displayed as a dotted black line. True spikes times obtained from an electrophysiological recording are displayed as black vertical lines.

2010; Pnevmatikakis et al., 2013; Theis et al., 2016; Deneux et al., 2016; Sasaki et al., 2008; Vogelstein et al., 2009; Yaksi and Friedrich, 2006; Vogelstein et al., 2010; Holekamp et al., 2008; Friedrich and Paninski, 2016; Friedrich et al., 2017). In this dissertation, we focus on the second step. Figure 2.1 displays an example fluorescence trace.

To infer the spike times, we revisit an auto-regressive model for calcium dynamics that has been considered by a number of authors (Vogelstein et al., 2010; Pnevmatikakis et al., 2016; Friedrich and Paninski, 2016; Friedrich et al., 2017). We closely follow the notation of Friedrich et al. (2017). This model posits that y_t, the fluorescence at time t, is a noisy realization of c_t, the unobserved underlying calcium concentration at the tth timestep. In the absence of a spike at the tth timestep ($z_t = 0$), the calcium concentration decays according to a first-order auto-regressive process. However, if a spike occurs at the tth timestep ($z_t > 0$),

then the calcium concentration increases. Thus,

$$\begin{aligned} y_t &= \beta_0 + \beta_1 c_t + \epsilon_t, \quad \epsilon_t \sim_{\text{ind.}} (0, \sigma^2), \quad t = 1, \ldots, T; \\ c_t &= \gamma c_{t-1} + z_t, \quad t = 2, \ldots, T. \end{aligned} \tag{2.1}$$

In (2.1), the quantity $0 < \gamma < 1$ is the parameter in the auto-regressive model. We further note that the quantity y_t in (2.1) is observed; all other quantities are unobserved. In Chapter 3 and Chapter 4 we show how (2.1) is used to estimate spike times $\{s : \hat{z}_s > 0\}$.

Chapter 3

EXACT SPIKE TRAIN INFERENCE VIA ℓ_0 OPTIMIZATION

This work is published in the *Annals of Applied Statistics* (Jewell and Witten, 2018).

3.1 Introduction

In Chapter 2, we described an experimental technique called calcium imaging to measure from large populations of neurons. For each neuron, this technique results in a time series of fluorescent intensities y_t that we model as a noisy observation of the underlying calcium concentration c_t. Recall the model for this process (2.1),

$$y_t = \beta_0 + \beta_1 c_t + \epsilon_t, \quad \epsilon_t \sim_{\text{ind.}} (0, \sigma^2), \quad t = 1, \ldots, T;$$
$$c_t = \gamma c_{t-1} + z_t, \quad t = 2, \ldots, T.$$

Since we would like to know whether a spike occurred at the tth timestep, the parameter of interest is z_t. Figure 3.1(a) displays a small dataset generated according to (2.1).

In what follows, for ease of exposition, we assume $\beta_0 = 0$ and $\beta_1 = 1$ in (2.1). This assumption is made without loss of generality, since β_0 and β_1 can be estimated from the data, and the observed fluorescence $y_1, \ldots, y_T$ centered and scaled accordingly.

Vogelstein et al. (2010), Friedrich and Paninski (2016), and Friedrich et al. (2017) seek to interpret z_t in (2.1) as the *number* of spikes at the tth timestep. Thus, in principle it would be desirable to use a count-valued distribution, such as the Poisson distribution, as a prior on z_t. However, because maximum a posteriori estimation of z_t in (2.1) using a Poisson distribution is computationally intractable, they instead suppose that z_t has an exponential distribution (Vogelstein et al., 2010). This leads Vogelstein et al. (2010) to the optimization

problem

$$\underset{c_1,\ldots,c_T,z_2,\ldots,z_T}{\text{minimize}} \left\{ \frac{1}{2}\sum_{t=1}^{T}(y_t - c_t)^2 + \lambda \sum_{t=2}^{T} |z_t| \right\} \text{ subject to } z_t = c_t - \gamma c_{t-1} \geq 0, \tag{3.1}$$

where λ is a nonnegative tuning parameter that controls the trade-off between the fit of the estimated calcium to the observed fluorescence, and the sparsity of the estimated spike vector $\hat{z}_2, \ldots, \hat{z}_T$. Friedrich and Paninski (2016) and Friedrich et al. (2017) instead consider a closely-related problem that results from including an additional ℓ_1 penalty for the initial calcium concentration,

$$\underset{c_1,\ldots,c_T,z_2,\ldots,z_T}{\text{minimize}} \left\{ \frac{1}{2}\sum_{t=1}^{T}(y_t - c_t)^2 + \lambda|c_1| + \lambda \sum_{t=2}^{T} |z_t| \right\} \text{ subject to } z_t = c_t - \gamma c_{t-1} \geq 0. \tag{3.2}$$

Both (3.1) and (3.2) are convex optimization problems, which can be solved for the global optimum using a well-developed set of optimization algorithms (Boyd and Vandenberghe, 2004; Hastie et al., 2009, 2015; Bien and Witten, 2016).

Because $\hat{z}_2, \ldots, \hat{z}_T$ are not integer-valued, they cannot be directly interpreted as the number of spikes at each timestep; however, informally, a larger value of $\hat{z}_t$ can be interpreted as indicating greater certainty that one or more spikes occurred at the tth timestep.

In this chapter, we re-consider the model (2.1) that originally motivated the optimization problems (3.1) and (3.2) in the recent literature (Vogelstein et al., 2010; Friedrich and Paninski, 2016; Friedrich et al., 2017). Rather than interpreting z_t in (2.1) as the number of spikes at the tth timestep, we interpret its sign as an indicator for whether or not at least one spike occurred: that is, $z_t = 0$ indicates no spikes at the tth timestep, and $z_t > 0$ indicates the occurrence of at least one spike. Then (2.1) leads naturally to the optimization problem

$$\underset{c_1,\ldots,c_T,z_2,\ldots,z_T}{\text{minimize}} \left\{ \frac{1}{2}\sum_{t=1}^{T}(y_t - c_t)^2 + \lambda \sum_{t=2}^{T} 1_{(z_t \neq 0)} \right\} \text{ subject to } z_t = c_t - \gamma c_{t-1} \geq 0, \tag{3.3}$$

where $1_{(A)}$ is an indicator variable that equals 1 if the event A holds, and 0 otherwise. In (3.3), λ is a non-negative tuning parameter that controls the trade-off between the fit of the estimated calcium to the observed fluorescence, and the number of timesteps at which a spike is estimated to occur.

Unfortunately, the optimization problem (3.3) is combinatorial and non-convex, due to the presence of the indicator variable. In the statistics literature, this term is known as an ℓ_0 penalty. It is well-known that optimization involving ℓ_0 penalties is typically computationally intractable: in general, no efficient algorithms are available to solve for the global optimum.

In fact, the convex optimization problem (3.1) considered in Vogelstein et al. (2010), and its close cousin (3.2) considered in Friedrich and Paninski (2016) and Friedrich et al. (2017), can be viewed as convex relaxations to the problem (3.3). That is, if we replace the ℓ_0 penalty in (3.3) with an ℓ_1 penalty, then that we arrive exactly at the problem (3.1).

3.1.1 Contribution of This Chapter

In the previous subsection, we established that the optimization problems (3.1) and (3.2), studied by Vogelstein et al. (2010), Friedrich and Paninski (2016), and Friedrich et al. (2017), can be seen as convex relaxations of the ℓ_0 optimization problem (3.3), which follows directly from the model (2.1). In fact, under the model (2.1), (3.3) is the "right" optimization problem to be solving; (3.1) and (3.2) are simply computationally tractable approximations to this problem. (In fact, Friedrich et al. (2017) allude to this in the "Hard shrinkage and ℓ_0 penalty" section of their paper.)

However, using an ℓ_1 norm to approximate an ℓ_0 norm comes with computational advantages at the expense of substantial performance disadvantages: in particular, the use of an ℓ_1 penalty tends to *overshrink* the fitted estimates (Zou, 2006). This can be seen quite clearly in Figures 3.1(b) and 3.1(c). Retaining only the four spikes in Figure 3.1(c) associated with the largest increases in calcium leads to an improvement in spike detection (Figure 3.1(e); this is referred to as the *post-thresholding ℓ_1 estimator* in what follows), but still one of the four true spikes is missed.

In this chapter, we consider a slight modification of (3.3) that results from removing the positivity constraint,

$$\underset{c_1,\ldots,c_T,z_2,\ldots,z_T}{\text{minimize}} \left\{ \frac{1}{2} \sum_{t=1}^{T} (y_t - c_t)^2 + \lambda \sum_{t=2}^{T} 1_{(z_t \neq 0)} \right\} \text{ subject to } z_t = c_t - \gamma c_{t-1}. \tag{3.4}$$

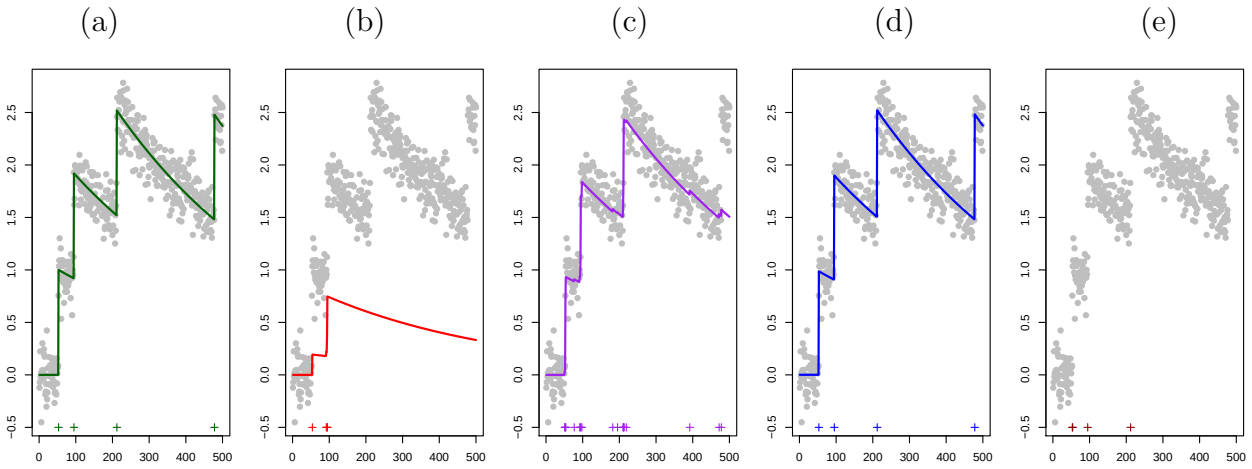


Figure 3.1: A toy simulated data example. In each panel, the x-axis represents time. Observed fluorescence values are displayed in (•). *(a)*: Unobserved calcium concentrations (—) and true spike times (+). Data were generated according to the model (2.1). *(b)*: Estimated calcium concentrations (—) and spike times (+) that result from solving the ℓ_1 optimization problem (3.2) with the value of λ that yields the true number of spikes. This value of λ leads to very poor estimation of both the underlying calcium dynamics and the spikes. *(c)*: Estimated calcium concentrations (—) and spike times (+) that result from solving the ℓ_1 optimization problem (3.2) with the largest value of λ that results in at least one estimated spike within the vicinity of each true spike. This value of λ results in 19 estimated spikes, which is far more than the true number of spikes. The poor performance of the ℓ_1 optimization problem in panels (b) and (c) is a consequence of the fact that the ℓ_1 penalty performs shrinkage as well as spike estimation; this is discussed further in Section 3.1.1. *(d)*: Estimated calcium concentrations (—) and spike times (+) that result from solving the ℓ_0 optimization problem (3.4). *(e)* The four spikes in panel (c) associated with the largest estimated increase in calcium (+); we refer to this in the text as the post-thresholding ℓ_1 estimator. Since the estimated calcium is not well-defined after post-thresholding, we do not plot the estimated calcium concentration.

In practice, the distinction between the problems (3.4) and (3.3) is quite minor: on real data applications, for appropriate choices of the decay rate γ, the solution to (3.4) tends to satisfy the constraint in (3.3), and so the solutions are identical.

Like problem (3.3), solving problem (3.4) for the global optimum appears, at a glance, to be computationally intractable — we (the authors) are only aware of a few ℓ_0 optimization problems for which exact solutions can be obtained via efficient algorithms!

However, in this chapter, we show that in fact, (3.4) is a rare ℓ_0 optimization problem that can be *exactly solved for the global optimum using an efficient algorithm*. This is because (3.4) can be seen as a changepoint detection problem, for which efficient algorithms that run in no more than $\mathcal{O}(T^2)$ time, and often closer to $\mathcal{O}(T)$ time, are available. Furthermore, our implementation of the exact algorithm for solving (3.4) yields excellent results relative to the convex approximation (3.2) considered by Friedrich and Paninski (2016) and Friedrich et al. (2017). This vastly improved performance can be seen in Figure 3.1(d).

The rest of this chapter is organized as follows. In Section 3.2, we present an exact algorithm for solving the ℓ_0 problem (3.4). In Section 3.3, we investigate the performance of this algorithm, relative to the algorithm of Friedrich and Paninski (2016) and Friedrich et al. (2017) for solving the ℓ_1 problem (3.2), in a simulation study. In Section 3.4, we investigate the performances of both algorithms for spike train inference on a data set for which the true spike times are known (Chen et al., 2013; GENIE Project, 2015) and on a data set from the Allen Brain Observatory (Allen Institute for Brain Science, 2016; Hawrylycz et al., 2016). Finally, we close with a discussion in Section 3.5. Technical details and additional results can be found in Appendix A.

3.2 An Exact Algorithm For Solving Problem (3.4)

In Section 3.2.1, we show that problem (3.4) can be viewed as a changepoint detection problem. In Sections 3.2.2 and 3.2.3, we apply existing algorithms for changepoint detection in order to efficiently solve (3.4) for the global optimum in $\mathcal{O}(T^2)$ and in substantially fewer than $O(T^2)$ operations, respectively.

Timing results are presented in Section 3.2.4. We discuss selection of the tuning parameter λ and auto-regressive parameter γ in (3.4) in Appendix A.3.

3.2.1 Recasting (3.4) *as a Changepoint Detection Problem*

Recall that our goal is to solve the ℓ_0 optimization problem (3.4), or equivalently, to compute $\hat{c}_1, \ldots, \hat{c}_T$ that solve the optimization problem

$$\underset{c_1,\ldots,c_T}{\text{minimize}} \left\{ \frac{1}{2} \sum_{t=1}^{T} (y_t - c_t)^2 + \lambda \sum_{t=2}^{T} 1_{(c_t - \gamma c_{t-1} \neq 0)} \right\}.$$

We estimate a spike event at the tth timestep if $\hat{c}_t \neq \gamma \hat{c}_{t-1}$. (We refer to this as a "spike event", rather than a spike, since $\hat{c}_t \neq \gamma \hat{c}_{t-1}$ indicates the presence of at least one spike at the tth timepoint, but does not directly provide an estimate of the number of spikes.) We now make two observations about this optimization problem.

1. Given that a spike event is estimated at the tth timestep, the estimated calcium concentration at any time $t_1 < t$ is independent of the estimated calcium concentration at any time $t_2 \geq t$.

2. Given that two spike events are estimated at the tth and t'th timesteps with $t < t'$, and no spike events are estimated in between the tth and t'th timesteps, the calcium concentration is estimated to decay exponentially between the tth and t'th timesteps.

This motivates us to consider the relationship between (3.4) and a *changepoint detection problem* (Aue and Horváth, 2013; Braun and Muller, 1998; Davis et al., 2006; Yao, 1988; Lee, 1995; Jackson et al., 2005; Killick et al., 2012; Maidstone et al., 2017b) of the form

$$\underset{0=\tau_0<\tau_1<\ldots<\tau_k<\tau_{k+1}=T,k}{\text{minimize}} \left\{ \sum_{j=0}^{k} \mathcal{D}(y_{(\tau_j+1):\tau_{j+1}}) + \lambda k \right\}, \tag{3.5}$$

where

$$\mathcal{D}(y_{a:b}) \equiv \min_{c_a, c_t = \gamma c_{t-1},\ t=a+1,\ldots,b} \left\{ \frac{1}{2} \sum_{t=a}^{b} (y_t - c_t)^2 \right\}. \tag{3.6}$$

In (3.5), we are simultaneously minimizing the objective over the times at which the changepoints $(\tau_1, \ldots, \tau_k)$ occur and the number of changepoints (k); the parameter λ controls the relative importance of these two terms.

The following result establishes an equivalence between (3.5) and (3.4).

Proposition 3.1 *There is a one-to-one correspondence between the set of estimated spike events in the solution to* (3.4) *and the set of changepoints* $0 = \tau_0, \tau_1, \ldots, \tau_k, \tau_{k+1} = T$ *in the solution to* (3.5)*, in the sense that* $\hat{c}_t \neq \gamma \hat{c}_{t-1}$ *if and only if* $t \in \{\tau_1 + 1, \ldots, \tau_k + 1\}$*. Furthermore, given the set of changepoints, the solution to* (3.4) *takes the form*

$$\hat{c}_t = \begin{cases} \gamma \hat{c}_{t-1} & \tau_j + 2 \leq t \leq \tau_{j+1} \\ \frac{\sum_{t=\tau_j+1}^{\tau_{j+1}} y_t \gamma^{t-(\tau_j+1)}}{\sum_{t=\tau_j+1}^{\tau_{j+1}} \gamma^{2(t-(\tau_j+1))}} & t = \tau_j + 1 \end{cases},$$

for $j \in \{0, \ldots, k\}$.

Proposition 3.1 indicates that in order to solve (3.4), it suffices to solve (3.5). (We note that due to a slight discrepancy between the conventions used in the changepoint detection literature and the notion of a spike event in this chapter, the indexing in Proposition 3.1 is a little bit awkward, in the sense that the kth spike event is estimated to occur at time $\tau_k + 1$, rather than at time τ_k.)

In the next two sections, we will make use of the following result.

Proposition 3.2 *The quantity* (3.6) *has a closed-form expression,*

$$\mathcal{D}(y_{a:b}) = \sum_{t=a}^{b} \frac{y_t^2}{2} - \mathcal{C}(y_{a:b}) \sum_{t=a}^{b} y_t \gamma^{t-a} + \frac{\mathcal{C}(y_{a:b})^2}{2} \sum_{t=a}^{b} \gamma^{2(t-a)}, \text{ where}$$
$$\mathcal{C}(y_{a:b}) = \frac{\sum_{t=a}^{b} y_t \gamma^{t-a}}{\sum_{t=a}^{b} \gamma^{2(t-a)}}.$$

Furthermore, given $\mathcal{D}(y_{a:b})$*, we can calculate* $\mathcal{D}(y_{a:(b+1)})$ *in constant time.*

Propositions 3.1 and 3.2 are proven in Appendix A.

3.2.2 An Algorithm For Solving (3.4) *In* $\mathcal{O}(T^2)$ *Operations*

In this section, we apply a dynamic programming algorithm proposed by Fisher (1958), Bellman (1961), Auger and Lawrence (1989), and Jackson et al. (2005) in order to solve the changepoint detection problem (3.5) for the global optimum in $\mathcal{O}(T^2)$ time. Due to the equivalence between (3.5) and (3.4) established in Proposition 3.1, this algorithm also solves problem (3.4).

Roughly speaking, this algorithm recasts the very difficult problem of choosing the times of all changepoints simultaneously into the much simpler problem of choosing the time of just the most recent changepoint. In greater detail, consider solving (3.5) on the first s timesteps. Define $F(0) \equiv -\lambda$, and for $s \geq 1$, define

$$\begin{aligned}
F(s) &= \min_{0=\tau_0<\tau_1<\cdots<\tau_k<\tau_{k+1}=s,k} \left\{ \sum_{j=0}^{k} \mathcal{D}(y_{(\tau_j+1):\tau_{j+1}}) + \lambda k \right\} \\
&= \min_{0=\tau_0<\tau_1<\cdots<\tau_k<\tau_{k+1}=s,k} \left\{ \sum_{j=0}^{k} \left[\mathcal{D}(y_{(\tau_j+1):\tau_{j+1}}) + \lambda \right] - \lambda \right\} \\
&= \min_{0=\tau_0<\tau_1<\cdots<\tau_k<\tau_{k+1}=s,k} \left\{ \sum_{j=0}^{k-1} \left[\mathcal{D}(y_{(\tau_j+1):\tau_{j+1}}) + \lambda \right] - \lambda + \mathcal{D}(y_{(\tau_k+1):\tau_{k+1}}) + \lambda \right\} \\
&= \min_{0\leq\tau_k<\tau_{k+1}=s} \left\{ \min_{0=\tau_0<\tau_1<\cdots<\tau_{k'}<\tau_{k'+1}=\tau_k,k'} \left\{ \sum_{j=0}^{k'} \left[\mathcal{D}(y_{(\tau_j+1):\tau_{j+1}}) + \lambda \right] - \lambda \right\} + \mathcal{D}(y_{(\tau_k+1):\tau_{k+1}}) + \lambda \right\} \\
&= \min_{0\leq\tau<s} \left\{ F(\tau) + \mathcal{D}(y_{(\tau+1):s}) + \lambda \right\}. \qquad (3.7)
\end{aligned}$$

In other words, in order to solve (3.5), we need simply identify the time of the most recent changepoint, and then solve (3.5) on all earlier times.

This recursion gives a simple recipe for evaluating $F(T)$ efficiently: set $F(0) = -\lambda$, and compute $F(1), F(2), \ldots, F(T)$ based on previously calculated (and stored) values. For example, at $s = 1$, calculate and store

$$F(1) = \min_{0\leq\tau<1} \left\{ F(\tau) + \mathcal{D}(y_{(\tau+1):1}) + \lambda \right\} = F(0) + \mathcal{D}(y_1) + \lambda,$$

and then at $s = 2$ use the previously calculated values $F(0)$ and $F(1)$ to compute the

minimum over a finite set with two elements

$$F(2) = \min_{\tau \in \{0,1\}} \left\{ F(\tau) + \mathcal{D}(y_{(\tau+1):2}) + \lambda \right\} = \min \left\{ F(0) + \mathcal{D}(y_{1:2}) + \lambda, F(1) + \mathcal{D}(y_2) + \lambda \right\}.$$

Given $F(1), \ldots, F(s-1)$, computing $F(s)$ requires minimizing over a finite set of size s, and therefore it has computational cost linear in s. The total cost of computing $F(T)$ is quadratic in the total number of timesteps, T, since there are $T+1$ subproblems: $\sum_{s=0}^{T} s = \mathcal{O}(T^2)$.

Full details are provided in Algorithm 3.1. We note that this algorithm is particularly efficient in light of Proposition 3.2, which makes it possible to perform a constant-time update to $\mathcal{D}(y_{(\tau+1):s})$ in order to compute $\mathcal{D}(y_{(\tau+1):(s+1)})$.

Algorithm 3.1: An $\mathcal{O}(T^2)$ Algorithm for Solving (3.4)

Initialize: $F(0) = -\lambda$, $cp(0) = \emptyset$

foreach $s = 1, 2, \ldots, T$ **do**

  Calculate $F(s) = \min_{0 \le \tau < s} \left\{ F(\tau) + \mathcal{D}(y_{(\tau+1):s}) + \lambda \right\}$

  Set $s' = \operatorname{argmin}_{0 \le \tau < s} \left\{ F(\tau) + \mathcal{D}(y_{(\tau+1):s}) + \lambda \right\}$

  Update $cp(s) = (cp(s'), s')$

end

Output : The number of spike events $k \equiv \text{card}(cp(T))$, the changepoints $\{\tau_1, \ldots, \tau_k\} \equiv cp(T)$, the spike times $\{\tau_1 + 1, \ldots, \tau_k + 1\}$, and the estimated calcium concentrations

$$\hat{c}_t \equiv \begin{cases} \gamma \hat{c}_{t-1} & \tau_j + 2 \le t \le \tau_{j+1} \\ \frac{\sum_{t=\tau_j+1}^{\tau_{j+1}} y_t \gamma^{t-(\tau_j+1)}}{\sum_{t=\tau_j+1}^{\tau_j+1} \gamma^{2(t-(\tau_j+1))}} & t = \tau_j + 1 \end{cases},$$

for $j = 0, \ldots, k$, where $\tau_0 = 0$.

3.2.3 Dramatic Speed-Ups Using Cost-Complexity Pruning

Killick et al. (2012) considered problems of the form (3.5) for which an assumption on $\mathcal{D}(\cdot)$

holds; this assumption is satisfied by (3.6). The main insight of their paper is as follows. Suppose that $s < r$ and $F(s) + \mathcal{D}(y_{(s+1):r}) > F(r)$. Then for any $q > r$, it is mathematically impossible for the most recent changepoint before the qth timestep to have occurred at the sth timestep. This allows us to *prune* the set of candidate changepoints that must be considered in each step of Algorithm 3.2, leading to drastic speed-ups. Details are provided in Algorithm 3.2, which solves (3.4) for the global optimum.

Killick et al. (2012) show that under a general data generative process the expected complexity of this algorithm is $\mathcal{O}(T)$. In particular, they assume that the parameters θ for each segment are iid, data points y are iid from density $f(y|\theta)$, and that the segment lengths $S_1 = \tau_1,\ S_2 = \tau_2 - \tau_1, \ldots$ are iid and independent of the segment parameters $\theta_1, \theta_2, \ldots$. Furthermore, they assume two weak technical conditions on the density $f(y|\theta)$, and two conditions on the segment length:

1. $\mathbb{E}(S_j^4) < \infty$;

2. $\mathbb{E}\left(\log f(y_t|\theta_j) - \log f(y_t|\theta^*)\right) > \frac{\lambda}{\mathbb{E}(S_j)}$, where θ_j is the true segment parameter at time t, and $\theta^* = \operatorname{argmax}\left\{\mathbb{E}\log f(y|\theta)\right\}$ maximizes the expected log-likelihood.

Importantly, $\mathbb{E}(S_j^4) < \infty$ implies that the expected number of changepoints increases linearly with the length of the data. This is reasonable in the context of calcium imaging data, in which we expect the number of neuron spike events to be linear in the length of the recording. See Theorem 3.2 of Killick et al. (2012) for additional details.

3.2.4 Timing Results For Solving (3.4)

We simulated data from (2.1) with $\gamma = 0.998$, $\sigma = 0.15$, and $z_t \sim_{\text{ind.}} \text{Poisson}(\theta)$ with $\theta \in \{0.1, 0.01, 0.001\}$. We solved (3.4) with $\lambda = 1$, using our `R`-language implementations of Algorithms 3.1 and 3.2.

Timing results, averaged over 50 simulated data sets, are displayed in Figure 3.2. As expected, the running time of Algorithm 3.1 scales quadratically in the length of the time

Algorithm 3.2: An Algorithm for Solving (3.4) in Substantially Fewer than $\mathcal{O}(T^2)$ Operations

Initialize: $F(0) = -\lambda$, $cp(0) = \emptyset$, $\mathcal{E}_1 = \{0\}$

foreach $s = 1, 2, \ldots, T$ **do**

  Calculate $F(s) = \min_{\tau \in \mathcal{E}_s} \left\{ F(\tau) + \mathcal{D}(y_{(\tau+1):s}) + \lambda \right\}$

  Set $s' = \operatorname{argmin}_{\tau \in \mathcal{E}_s} \left\{ F(\tau) + \mathcal{D}(y_{(\tau+1):s}) + \lambda \right\}$

  Update $\mathcal{E}_{s+1} = \{\tau \in \{\mathcal{E}_s \cup s\} : F(\tau) + \mathcal{D}(y_{(\tau+1):s}) < F(s)\}$

  Update $cp(s) = (cp(s'), s')$

end

Output : The number of spike events $k \equiv \text{card}(cp(T))$, the changepoints $\{\tau_1, \ldots, \tau_k\} \equiv cp(T)$, the spike times $\{\tau_1 + 1, \ldots, \tau_k + 1\}$, and the estimated calcium concentrations

$$\hat{c}_t \equiv \begin{cases} \gamma \hat{c}_{t-1} & \tau_j + 2 \leq t \leq \tau_{j+1} \\ \frac{\sum_{t=\tau_j+1}^{\tau_{j+1}} y_t \gamma^{t-(\tau_j+1)}}{\sum_{t=\tau_j+1}^{\tau_{j+1}} \gamma^{2(t-(\tau_j+1))}} & t = \tau_j + 1 \end{cases},$$

for $j = 0, \ldots, k$, where $\tau_0 = 0$.

series, whereas the running time of Algorithm 3.2 is upper bounded by that of Algorithm 3.1. Furthermore, the running time of Algorithm 3.2 decreases as the firing rate increases. The Chen et al. (2013) dataset explored in Section 3.4.1 has firing rate on the same order of magnitude as the middle panel, $\theta = 0.01$. Using Algorithm 3.2, we can solve (3.4) for the global optimum in a few minutes on a 2.5 GHz Intel Core i7 Macbook Pro on fluorescence traces of length $100,000$ with moderate to high firing rates.

We note here that Algorithm 3.2 for solving (3.4) is much slower than the algorithm of Friedrich et al. (2017) for solving (3.2), which is implemented in Cython and has approximately linear running time. In Chapter 4, we develop a faster algorithm for solving (3.4) using ideas from Johnson (2013), Maidstone et al. (2017b), and Hocking et al. (2017). Fur-

thermore, a much faster implementation of Algorithm 3.2 would be possible using a language other than `R`.

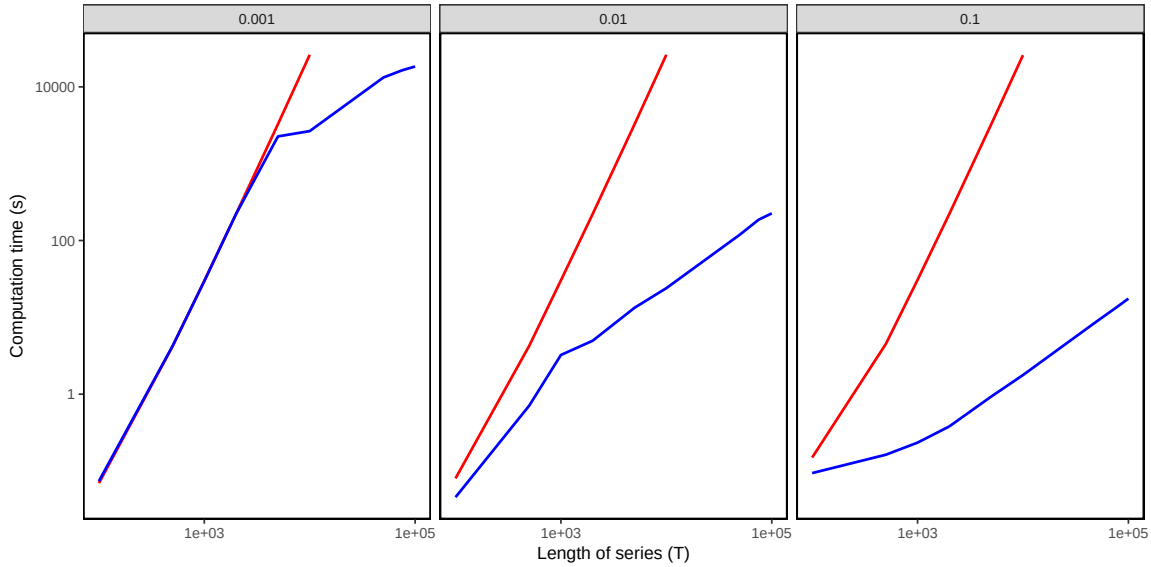


Figure 3.2: Timing results for solving (3.4) for the global optimum, using Algorithms 3.1 (—) and 3.2 (—). The x-axis displays the length of the time series (T), and the y-axis displays the average running time, in seconds. Each panel, from left to right, corresponds to data simulated according to (2.1) with $z_t \sim_{\text{i.i.d.}} \text{Poisson}(\theta)$, with $\theta \in \{0.001, 0.01, 0.1\}$. Standard errors are on average $< 0.1\%$ of the mean compute time. Additional details are provided in Section 3.2.4.

3.3 Simulation Study

3.3.1 Comparison Methods

In this section, we use *in silico* data to demonstrate the performance advantages of the ℓ_0 approach (3.4) over two competing approaches:

1. The ℓ_1 proposal (3.2) of Friedrich and Paninski (2016) and Friedrich et al. (2017),

which involves a single tuning parameter λ.

2. A thresholded version of the ℓ_1 estimator. Letting $\hat{z}_2, \ldots, \hat{z}_T$ denote the solution to (3.2), we define the *post-thresholding estimator* as

$$\tilde{z}_t = \hat{z}_t 1_{(\hat{z}_t \geq L)}, \quad t = 2, \ldots, T, \tag{3.8}$$

for L a positive constant. In other words, the post-thresholding estimator retains only the estimated spikes for which the estimated increase in calcium exceeds a threshold L. The post-thresholding estimator involves two tuning parameters: λ in (3.2), as well as the value of L used to perform thresholding.

The post-thresholding estimator is motivated by the fact that the solution to (3.2) tends to yield many "small" spikes: i.e. $\hat{z}_t$ is near zero, but not exactly equal to zero, for many timesteps. In fact, this can be seen in Figure 3.1(c). As seen in Figure 3.1(e), the post-thresholding estimator has the potential to improve the performance of the ℓ_1 estimator by removing some of these small spikes. Of course, the post-thresholding estimator with $L = 0$ is identical to the ℓ_1 estimator from (3.2).

3.3.2 Performance Measures

We measure performance of each method based on two criteria: (i) error in calcium estimation, and (ii) error in spike detection.

We consider the mean of squared differences between the true calcium concentration in (2.1) and the estimated calcium concentration that solves (3.4),

$$\text{MSE}(c, \hat{c}) = \frac{1}{T} \sum_{t=1}^{T} (c_t - \hat{c}_t)^2. \tag{3.9}$$

This quantity involves the unobserved calcium concentrations, $c_1, \ldots, c_T$, and thus can only be computed on simulated data. Furthermore, this quantity can be computed for our ℓ_0 proposal (3.4) and for the ℓ_1 proposal (3.2), but not for the post-thresholding estimator

(3.8), since the post-thresholding estimator does not lead to an estimate of the underlying calcium concentrations.

We now consider the task of quantifying the error in spike detection. We make use of the Victor-Purpura distance metric (Victor and Purpura, 1996, 1997), which defines the distance between two spike trains as the minimum cost of transforming one spike train to the other through spike insertion, deletion, or translation. We also use the van Rossum distance (van Rossum, 2001), defined as the mean squared difference between two spike trains that have been convolved with an exponential kernel with timescale $\tau = 2$.

3.3.3 Results

We generated 100 simulated data sets according to (2.1) with parameter settings $\gamma = 0.96$, $T = 5000$, $\sigma = 0.15$, and $z_t \sim_{\text{i.i.d.}} \text{Poisson}(0.01)$.

On each simulated data set, we solved (3.4) and (3.2) for a range of values of the tuning parameter λ. Moreover, we post-thresholded the ℓ_1 solution, as in (3.8), with five different threshold values: $L \in \{0, 0.125, 0.250, 0.375, 0.500\}$.

Figure 3.3(a) displays the error in spike event detection for the van Rossum distance, Figure 3.3(b) displays the error in spike event detection for the Victor-Purpura distance metric, and Figure 3.3(c) displays the error in calcium estimation (3.9), for the ℓ_0 problem (3.4) and the ℓ_1 problem (3.2), for a range of values of λ. Results are averaged over the 50 simulated data sets.

As mentioned earlier, since the calcium concentration is not defined for the post-thresholding estimator (3.8), the post-thresholding estimator is not displayed in Figure 3.3(c). In Figures 3.3(a) and 3.3(b), five distinct curves are displayed for the post-thresholding operator; each corresponds to a distinct value of L. Note that as L increases, the maximum possible number of estimated spikes from the post-thresholding estimator decreases. For example, with $\lambda = 0$ and $L = 0.5$, no more than approximately 50 spikes are estimated by the post-thresholding estimator. For this reason, some of the curves corresponding to the post-thresholding estimator appear truncated in Figures 3.3(a) and 3.3(b).

Figure 3.3 reveals that the ℓ_0 estimator (3.4) results in dramatically lower errors in both calcium estimation and spike detection than the ℓ_1 estimator (3.2) (which is equivalent to the post-thresholding operator with $L = 0$). Although post-thresholding with $L > 0$ improves upon the unthresholded ℓ_1 estimator, the ℓ_0 estimator still substantially outperforms all competitors in Figures 3.3(a) and 3.3(b). Moreover, the ℓ_0 estimator requires just a single tuning parameter λ in (3.4), whereas the post-thresholding procedure involves two tuning parameters, λ in (3.2) and L in (3.8), leading to challenges in tuning parameter selection.

Furthermore, the ℓ_0 problem (3.4) achieves the lowest errors in both calcium estimation and spike detection when applied using a value of the tuning parameter λ that yields approximately 50 estimated spikes, which is the expected number of spikes in this simulation. This suggests that it should be possible to use a cross-validation scheme to select the tuning parameter λ for the ℓ_0 approach.

By contrast, in Figure 3.3(b), the ℓ_1 approach achieves its lowest error in calcium estimation when far more than 50 spikes are estimated. This is a consequence of the fact that the ℓ_1 penalty simultaneously reduces the number of estimated spikes and shrinks the estimated calcium. Therefore, the value of the tuning parameter λ in (3.2) that yields the most accurate estimate of calcium will result in severe over-estimation of the number of spikes. This means that a similar cross-validation scheme will not perform well for the ℓ_1 approach.

3.4 Application To Calcium Imaging Data

In this section, we apply our ℓ_0 proposal (3.4) and the ℓ_1 proposal of Friedrich and Paninski (2016) and Friedrich et al. (2017) (3.2), both with and without post-thresholding (3.8), to two calcium imaging data sets. In the first data set, the true spike times are known (Chen et al., 2013; GENIE Project, 2015), and so we can directly assess the spike detection accuracy of each proposal. In the second data set, the true spike times are unknown (Allen Institute for Brain Science, 2016; Hawrylycz et al., 2016); nonetheless, we are able to make a qualitative comparison of the results of the ℓ_1 and ℓ_0 proposals.

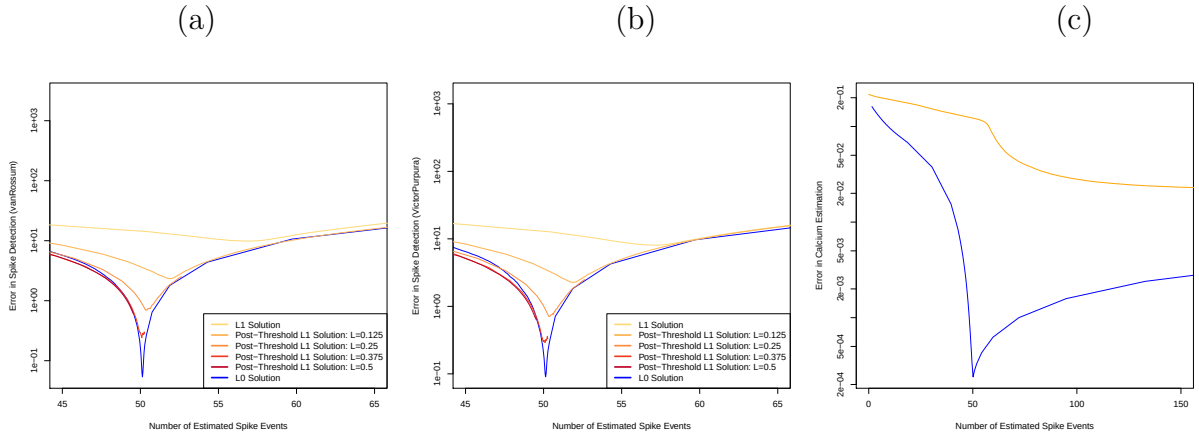


Figure 3.3: Simulation study to assess the error in spike detection and calcium estimation, for the ℓ_1 (3.2), post-thresholded ℓ_1 (3.8), and ℓ_0 (3.3) problems. *(a)*: Error in spike detection, measured using van Rossum distance. *(b)*: Error in spike detection, measured using Victor-Purpura distance. *(c)*: Error in calcium estimation (3.9). Simulation details are provided in Section 3.3.

3.4.1 Application to Chen et al. (2013) Data

We first consider a data set that consists of simultaneous calcium imaging and electrophysiological measurements (Chen et al., 2013; GENIE Project, 2015), obtained from the Collaborative Research in Computational Neuroscience portal (`http://crcns.org/data-sets/methods/cai-1/about-cai-1`). In what follows, we refer to the spike times inferred from the electrophysiological measurements as the "true" spikes.

The top panel of Figure 3.4 shows a 40-second recording from cell 2002, which expresses GCaMP6s. The data are measured at 60 Hz, for a total of 2400 timesteps. The raw fluorescence traces are DF/F transformed with a 20% percentile filter as in Figure 3 of Friedrich et al. (2017). In this 40-second recording, there are a total of 23 true spikes; therefore, we solved the ℓ_0 and ℓ_1 problems with $\gamma = 0.9864405$ using values of λ in (3.4) and (3.2) that yield 23 estimated spikes. Additionally, we solved the ℓ_1 problem with $\lambda = 1$, and post-

thresholded it according to (3.8) using $L = 0$, 0.1, and 0.13; these threshold values yielded 230, 54, and 23 estimated spikes, respectively.

Figure 3.4 displays the estimated spikes resulting from the ℓ_0 proposal, the estimated spikes resulting from the ℓ_1 proposal, the estimated spikes from post-thresholding the ℓ_1 solutions, and the ground truth spikes. We see that the ℓ_0 proposal has one false negative (i.e. it misses one true spike at around 7 seconds) and one false positive (i.e. it estimates a spike at around 36 seconds, where there is no true spike). By contrast, the ℓ_1 problem concentrates the 23 estimated spikes at three points in time, and therefore suffers from a substantial number of false positives as well as false negatives. Because the ℓ_1 penalty controls both the number of spikes and the estimated calcium, the ℓ_1 problem tends to put a large number of spikes in a row, each of which is associated with a very modest increase in calcium. This is consistent with the results seen in Figures 3.1 and 3.3. Post-thresholding the ℓ_1 estimator does lead to an improvement in results relative to the unthresholded ℓ_1 method; however, the post-thresholded solution with 23 spikes still tends to estimate a number of spikes in short succession when in fact only one true spike is present, and also misses several true spike events.

We note that the ℓ_0 method tends to estimate spike times one or two timesteps ahead of the true spike times. This is due to model misspecification: model (2.1) assumes that the calcium concentration increases instantaneously due to a spike event, and subsequently decays; however, we see from Figure 3.4 that in reality, a spike event is followed by an increase in calcium over the course of a few timesteps, before the onset of exponential decay. In practice, estimated spike times from the ℓ_0 method can be adjusted to account for this empirical observation.

In Appendix A.4, we apply an approach proposed by Friedrich et al. (2017) to approximate the solution to a non-convex problem using a greedy algorithm. This alternative approach performs quite a bit better than solving the ℓ_1 problem (3.2); however, it does not achieve the global optimum.

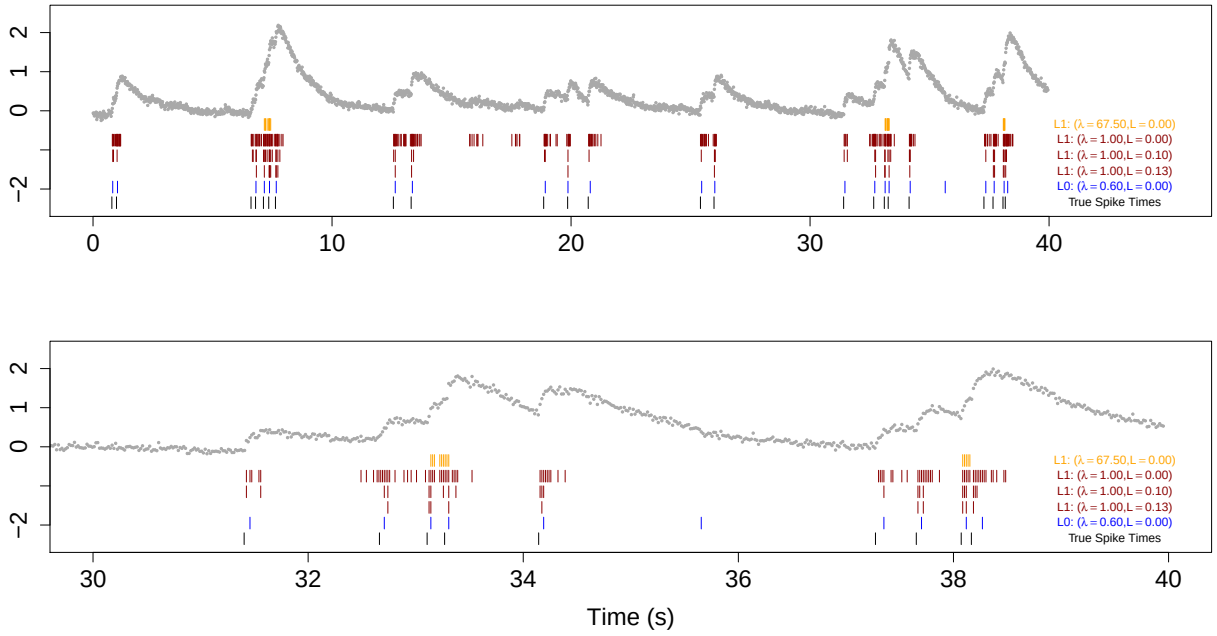


Figure 3.4: Spike detection for cell 2002 of the Chen et al. (2013) data. The observed fluorescence (•) and true spikes (—) are displayed. Estimated spike times from the ℓ_0 problem (3.3) are shown in (—), estimated spike times from the ℓ_1 problem (3.2) are shown in (—), and estimated spike times from the post-thresholding estimator (3.8) are shown in (—). Times $0s - 35s$ are shown in the top row; the second row zooms into time $30s - 40s$ in order to illustrate the behavior around a large increase in calcium concentration.

3.4.2 Application to Allen Brain Observatory Data

We now consider a data set from the Allen Brain Observatory, a large open-source repository of calcium imaging data from the mouse visual cortex (Allen Institute for Brain Science, 2016; Hawrylycz et al., 2016). For this data, the true spike times are not available, and so it is difficult to objectively assess the performances of the ℓ_1, post-thresholded ℓ_1, and ℓ_0 methods. Instead, for each method we present several fits that differ in the number of detected spikes. We argue that the ℓ_0 problem yields results that are qualitatively superior to those of the

competitors, in the sense that they are better supported by visual inspection of the data.

For the second ROI in NWB 510221121, we applied the ℓ_1, post-thresholded ℓ_1, and ℓ_0 methods to the first $10,000$ timesteps of the DF/F-transformed fluorescence traces. Since the data are measured at 30 Hz, this amounts to the first 333 seconds of the recording. Figure 3.5 shows the results obtained with $\gamma = 0.981756$. For the ℓ_0 and ℓ_1 estimators, we chose the values of λ in (3.2) and (3.4) in order to obtain 27, 49, and 128 estimated spikes. For the post-thresholded estimator (3.8), we set $\lambda = 1$, and then selected L to yield 27, 49, and 128 estimated spikes.

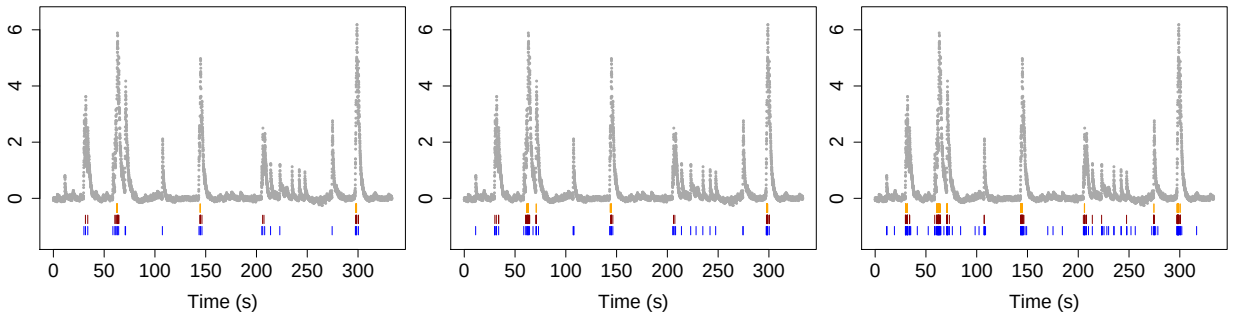
As in the previous subsection, we see that when faced with a large increase in fluorescence, the ℓ_1 problem tends to estimate a very large number of spikes in quick succession. For example, when 27 spikes are estimated, the ℓ_1 problem concentrates the estimated spikes at three points in time (Figure 3.5(a)). Even when 128 spikes are estimated, the ℓ_1 problem still seems to miss all but the largest peaks in the fluorescence data (Figure 3.5(c)). Post-thresholding the ℓ_1 estimator improves upon this issue somewhat, but spikes corresponding to smaller increases in fluorescence are still missed; this issue can be clearly seen in Figures 3.5(d)–(f), which zoom in on a smaller time window.

By contrast, the ℓ_0 problem can assign an arbitrarily large increase in calcium to a single spike event. Therefore, it seems to capture most of the visible peaks in the fluorescence data when 49 spikes are estimated (Figures 3.5(b) and 3.5(e)), and it captures all of them when 128 spikes are estimated (Figures 3.5(c) and 3.5(f)).

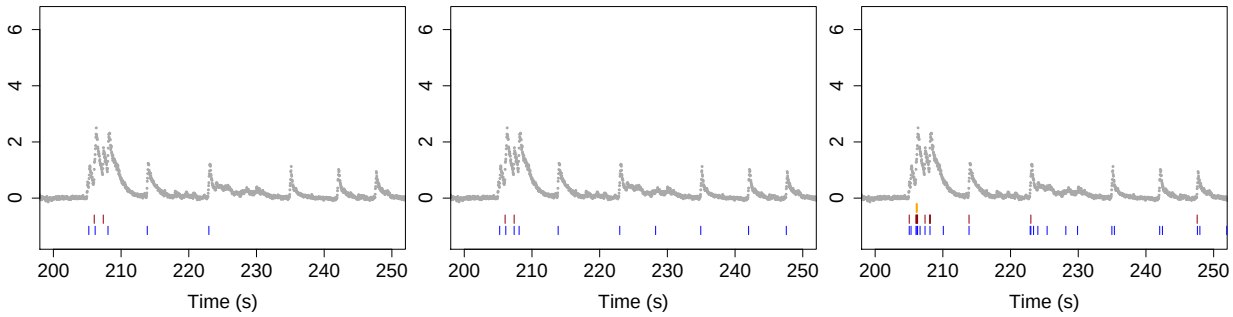
Though the true spike times are unknown for this data, based on visual inspection, the results for the ℓ_0 proposal seem superior to those of the ℓ_1 and post-thresholded ℓ_1 proposals.

3.5 Discussion

In this chapter, we considered solving the seemingly intractable ℓ_0 optimization problem (3.4) corresponding to the model (2.1). By recasting (3.4) as a changepoint detection problem, we were able to derive an algorithm to solve (3.4) for the global optimum in expected linear time.



(a) 27 spikes. (b) 49 spikes. (c) 128 spikes.



(d) Zoomed 27 spikes. (e) Zoomed 49 spikes. (f) Zoomed 128 spikes.

Figure 3.5: The first 10,000 timesteps from the second ROI in NWB 510221121 from the Allen Brain Observatory. Each panel displays the DF/F-transformed fluorescence (•), the estimated spikes from the ℓ_0 problem (—) (3.4), the estimated spikes from the ℓ_1 problem (—) (3.2), and the estimated spikes from post-thresholding the ℓ_1 problem (—) (3.8). The panels display results from applying the ℓ_1 and ℓ_0 methods with tuning parameter λ chosen to yield *(a)*: 27 spikes for each method; *(b)*: 49 spikes for each method; and *(c)*: 128 spikes for each method. The post-thresholding estimator was obtained by applying the ℓ_1 method with $\lambda = 1$, and thresholding the result to obtain 27, 49, or 128 spikes. *(d)–(f)*: As in (a)–(c), but zoomed in on 200-250 seconds.

We have shown in this chapter that solving the ℓ_0 optimization problem (3.4) leads to more accurate spike event detection than solving the ℓ_1 optimization problem (3.2) proposed by Friedrich et al. (2017). Indeed, this finding is intuitive: the ℓ_1 penalty and positivity constraint in (3.2) serves as a exponential prior on the increase in calcium at any given time point, and thereby effectively limits the amount that calcium can increase in response to a spike event. By contrast, the ℓ_0 penalty in (3.4) is completely agnostic to the amount by which a spike event increases the level of calcium. Consequently, it can allow for an arbitrarily large (or small) increase in fluorescence as a result of a spike event.

While approximations to the solution to the ℓ_0 problem (3.4) are possible (de Rooi and Eilers, 2011; de Rooi et al., 2014; Hugelier et al., 2016; Scott and Knott, 1974; Olshen et al., 2004; Fryzlewicz et al., 2014; Friedrich et al., 2017), there is no guarantee that such approaches will yield an attractive local optimum on a given data set. In this chapter, we completely bypass this concern by solving the ℓ_0 problem for the global optimum.

In this chapter, we have focused on the empirical benefits of the ℓ_0 problem (3.4) over the ℓ_1 problem (3.2). However, it is natural to wonder whether these empirical benefits are backed by statistical theory. Conveniently, both the ℓ_0 and ℓ_1 optimization problems are very closely-related to problems that have been well-studied in the statistical literature from a theoretical standpoint. In particular, in the special case of $\gamma = 1$, the ℓ_0 problem (3.4) was extensively studied in Yao and Au (1989) and Boysen et al. (2009). Furthermore, when $\gamma = 1$, the ℓ_1 problem (3.4) is very closely-related to the *fused lasso* optimization problem,

$$\underset{c_1,\ldots,c_T}{\text{minimize}} \left\{ \frac{1}{2} \sum_{t=1}^{T} (y_t - c_t)^2 + \lambda \sum_{t=2}^{T} |c_t - c_{t-1}| \right\},$$

which has also been extremely well-studied (Tibshirani et al., 2005; Mammen et al., 1997; Davies and Kovac, 2001; Harchaoui and Lévy-Leduc, 2010; Qian and Jia, 2012; Rojas and Wahlberg, 2014; Lin et al., 2016; Dalalyan et al., 2017). However, we leave a formal theoretical analysis of the relative merits of (3.4) and (3.2), in terms of ℓ_2 error bounds and spike recovery properties, to future work.

Our `R`-language software for our proposal is available on `CRAN` in the package

`LZeroSpikeInference`. Instructions for running this software in `python` can be found at `https://github.com/jewellsean/LZeroSpikeInference`.

Chapter 4

FAST NONCONVEX DECONVOLUTION OF CALCIUM IMAGING DATA

This work is published in *Biostatistics* (Jewell et al., 2019b).

4.1 Introduction

In Chapter 3, we showed that it is possible to efficiently solve the nonconvex optimization problem

$$\underset{c_1,\ldots,c_T,z_2,\ldots,z_T}{\text{minimize}} \left\{ \frac{1}{2}\sum_{t=1}^{T}(y_t - c_t)^2 + \lambda \sum_{t=2}^{T} 1_{(z_t \neq 0)} \right\} \text{ subject to } z_t = c_t - \gamma c_{t-1},$$

obtained by removing the positivity constraint, $c_t - \gamma c_{t-1} \geq 0$, from (3.3). The positivity constraint enforces the biological property that a firing neuron can only cause the calcium concentration to increase. Nonetheless, despite the slight loss in physical interpretability caused by the omission of the positivity constraint, in Chapter 3 we showed that solving (3.4) leads to improved performance over existing deconvolution approaches that perform a convex relaxation of (3.3). In particular, the method of Chapter 3 provides an accurate estimate of the *specific timesteps at which a neuron fires*.

Unfortunately, the algorithm proposed in Chapter 3 for solving (3.4) is too slow to conveniently run on large-scale data. For traces of 100,000 timesteps, the implementation runs in a few minutes for a single value of the tuning parameter λ; in practice the user must apply the algorithm over a fine grid of values of λ, leading potentially to hours of computation time for a single trace. Furthermore, a single experiment could result in hundreds or thousands of fluorescence traces (Ahrens et al., 2013; Vladimirov et al., 2014).

In this chapter, we develop a fast algorithm for solving problem (3.4); for traces of 100,000 timesteps our implementation runs in less than a second. Furthermore, this new algorithm

can easily accommodate the positivity constraint that was omitted from (3.4); in other words, we can directly solve problem (3.3). Additionally, we exploit ideas from Haynes et al. (2017) to efficiently "choose" good values of λ; that is, values of λ where the solution to (3.4) changes.

The algorithm we develop to solve (3.3) was used to obtain the key scientific results in the Allen Institute's main scientific paper from the Allen Brain Observatory (de Vries et al., 2020). Additionally, the Allen Institute for Brain Science recently released an update to their software development kit that provides users with the output from our algorithm for close to 60,000 neurons during different experimental conditions.

In what follows, we introduce our new algorithm for solving (3.3) and (3.4) in Section 4.2. We compare its performance in Section 4.3 to a convex relaxation of (3.3) on a number of calcium imaging datasets that were recently released as part of the `spikefinder` challenge (`http://spikefinder.codeneuro.org/`). We close with a discussion in Section 4.4.

4.2 A Fast Functional Pruning Algorithm For Solving Problems (3.4) *And* (3.3)

4.2.1 A Review Of Chapter 3

In Chapter 3, we point out that the ℓ_0 optimization problem (3.4) is equivalent to the changepoint problem,

$$\underset{0=\tau_0<\tau_1<\ldots<\tau_k<\tau_{k+1}=T,k}{\text{minimize}} \left\{ \sum_{j=0}^{k} \mathcal{D}(y_{(\tau_j+1):\tau_{j+1}}) + \lambda k \right\},$$

where

$$\mathcal{D}(y_{a:b}) \equiv \min_{\alpha} \left\{ \frac{1}{2} \sum_{t=a}^{b} \left(y_t - \alpha\gamma^{t-b}\right)^2 \right\}. \tag{4.1}$$

Here, we select the optimal changepoints $\tau_1, \ldots, \tau_k$ and the number of changepoints k such that the cost of segmenting the data into $k+1$ exponentially decaying regions is minimal, where (4.1) is the cost associated with the region that spans the ath to bth timesteps. (Note that (4.1) is a reparameterization of (3.6).)

Problems (3.5) and (3.4) are equivalent in the sense that $\hat{z}_{\hat{\tau}_1+1} \neq 0, \ldots, \hat{z}_{\hat{\tau}_k+1} \neq 0$ and all other $\hat{z}_t = 0$.

To solve the changepoint problem, in Chapter 3 we exploit a simple recursion (Fisher, 1958; Bellman, 1961; Jackson et al., 2005),

$$F(s) = \min_{0=\tau_0<\tau_1<\cdots<\tau_k<\tau_{k+1}=s,k} \left\{ \sum_{j=0}^{k} \mathcal{D}(y_{(\tau_j+1):\tau_{j+1}}) + \lambda k \right\} = \min_{0\leq\tau<s} \left\{ F(\tau) + \mathcal{D}(y_{(\tau+1):s}) + \lambda \right\}, \tag{4.2}$$

where $F(s)$ is the optimal cost of segmenting the data $y_{1:s} \equiv [y_1, \ldots, y_s]$, and where we define $F(0) \equiv -\lambda$. This results in an algorithm with computational complexity $\mathcal{O}(T^2)$, which can be substantially improved by noticing that the minimization on the right hand side of (4.2) can be performed over a smaller set $\mathcal{E}_s$ without sacrificing the global optimum (Killick et al., 2012). This algorithm runs in a few minutes for traces of length 100,000, and yields the global optimum to (3.4). We note that the recursion (4.2) does not naturally lead to an algorithm to solve (3.3); this is discussed in further detail in Section 4.2.3.

4.2.2 Functional Pruning For Solving (3.4)

Motivation For Functional Pruning

In order to motivate the potential for a much faster algorithm for solving (3.4) than the one proposed in Chapter 3, consider Figure 4.1.

In this figure, we are interested in determining the optimal cost of segmenting the data up to time 40, that is, calculating $F(40)$ in (4.2). Instead of directly applying the recursion (4.2), we consider a slightly different question: What is the optimal most recent changepoint before the 40th timestep, conditional on, the unknown calcium concentration c_{40}? Given the previously stored values $F(0), \ldots, F(39)$, and the data $y_{1:40}$, it is straightforward to calculate the best most recent changepoint $\tau^*(c_{40})$, as $\tau^*(c_{40}) = \underset{0\leq\tau<40}{\operatorname{argmin}} \left\{ F(\tau) + \frac{1}{2} \sum_{t=\tau+1}^{40} (y_t - \gamma^{t-40} c_{40})^2 + \lambda \right\}$, for any value of the calcium concentration c_{40}.

Figure 4.1 displays the most recent changepoint $\tau^*(c_{40})$ as a function of c_{40}. We observe

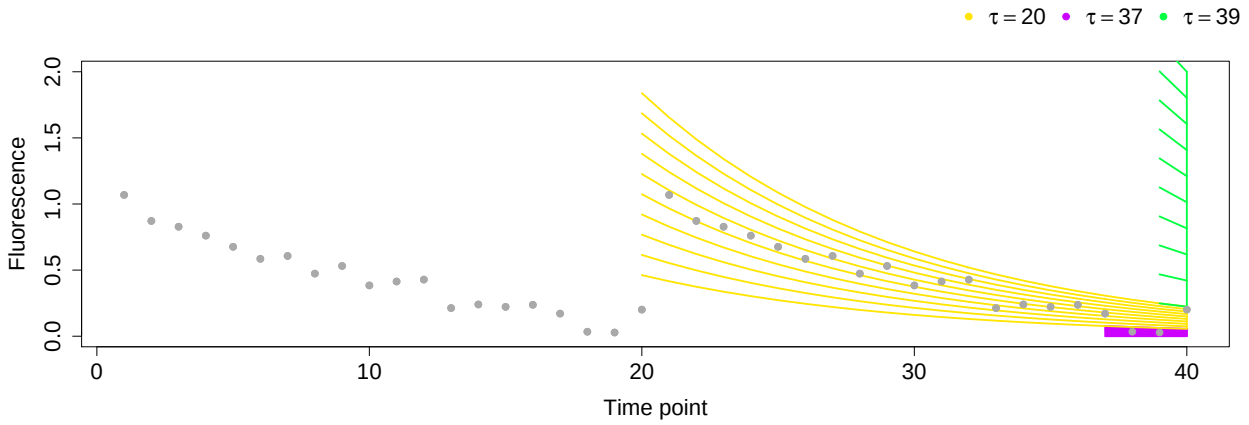


Figure 4.1: A simple example to show that there are only a few possible values for the most recent changepoint before timestep 40. We consider solving for the most recent changepoint, given data $y_{1:40}$, for each possible value of the calcium concentration at the 40th timestep, c_{40}. For each possible value of c_{40}, we display the estimated calcium concentration going back in time to the most recent changepoint before timestep 40. The colors indicate the time of the most recent changepoint. In this example, there are only three possibilities for the most recent changepoint: $\{20, 37, 39\}$. For example, $\tau^*(0.001) = 37, \tau^*(0.02) = 20$, and $\tau^*(1) = 39$.

that regardless of the value of the calcium at the current timestep — and consequently, regardless of the fluorescence values $y_{41}, y_{42}, y_{43}, \ldots, y_T$ — *the only possible times for the most recent changepoint before the 40th timestep are 20, 37, and 39*; that is, $\tau^*(c_{40}) \in \{20, 37, 39\}$ for all possible c_{40}.

However, the algorithm proposed in Chapter 3 does not exploit the fact that 20, 37, and 39 are the only possible times for the most recent changepoint before the 40th timestep: the minimization in (4.2) is performed over the set $\{0, \ldots, 39\}$, or else over a slightly smaller set $\{18, \ldots, 39\}$ using ideas from Killick et al. (2012). This suggests that by viewing the cost of segmenting the data up until the sth timestep as a function of the *calcium at the sth timestep*,

we could potentially develop an algorithm that is much faster than the one in Chapter 3 in that it would only require performing the minimization in (4.2) over $\{20, 37, 39\}$. The idea of using this type of conditioning was first suggested by Rigaill (2015) and Maidstone et al. (2017b), albeit to speed up algorithms for detecting changepoints in a different class of models.

The Functional Pruning Algorithm

To begin, we substitute the cost function $\mathcal{D}(y_{(\tau+1):s})$ into the recursion (4.2), in order to obtain

$$\begin{aligned}
F(s) &= \min_{0\leq\tau<s}\left\{F(\tau)+\mathcal{D}(y_{(\tau+1):s})+\lambda\right\} \\
&= \min_{0\leq\tau<s}\left\{F(\tau)+\min_{\alpha}\left\{\frac{1}{2}\sum_{t=\tau+1}^{s}\left(y_t-\alpha\gamma^{t-s}\right)^2\right\}+\lambda\right\} \\
&= \min_{\alpha}\min_{0\leq\tau<s}\left\{F(\tau)+\left\{\frac{1}{2}\sum_{t=\tau+1}^{s}\left(y_t-\alpha\gamma^{t-s}\right)^2\right\}+\lambda\right\} \\
&= \min_{\alpha}\min_{0\leq\tau<s}\text{Cost}_s^{\tau}(\alpha) \\
&= \min_{\alpha}\text{Cost}_s^{*}(\alpha), \qquad (4.3)
\end{aligned}$$

where

$$\text{Cost}_s^{\tau}(\alpha)\equiv F(\tau)+\frac{1}{2}\sum_{t=\tau+1}^{s}\left(y_t-\alpha\gamma^{t-s}\right)^2+\lambda, \qquad (4.4)$$

and

$$\text{Cost}_s^{*}(\alpha)=\min_{0\leq\tau<s}\text{Cost}_s^{\tau}(\alpha). \qquad (4.5)$$

In words, $\text{Cost}_s^{\tau}(\alpha)$ is the cost of partitioning the data up until time s, given that the most recent changepoint was at time τ, and the calcium at the sth timestep equals α. $\text{Cost}_s^{*}(\alpha)$ is the optimal cost of partitioning the data up until time s, given that the calcium at the sth timestep equals α.

The following proposition will prove useful in what follows.

Proposition 4.1 *For* $\text{Cost}_s^*(\alpha)$ *defined in* (4.5)*, the following recursion holds:*

$$\text{Cost}_s^*(\alpha) = \min\left\{\text{Cost}_{s-1}^*(\alpha/\gamma), \min_{\alpha'} \text{Cost}_{s-1}^*(\alpha') + \lambda\right\} + \frac{1}{2}(y_s - \alpha)^2. \tag{4.6}$$

The proof of Proposition 4.1 is in Appendix B.1. The recursion in (4.6) encompasses two possibilities: either there is a changepoint at the $(s-1)$st timestep, and we must determine the optimal cost up to that time, $\min_{\alpha'} \text{Cost}_{s-1}^*(\alpha') + \lambda + \frac{1}{2}(y_s - \alpha)^2$, or there is no changepoint at the $(s-1)$st timestep, $\text{Cost}_{s-1}^*(\alpha/\gamma) + \frac{1}{2}(y_s - \alpha)^2$. The recursion in (4.6) is reminiscent of (4.2), and raises the following question: can we use (4.6) as the basis for a recursive algorithm for solving the problem of interest, (3.4)? At first, it appears almost hopeless, since the recursion (4.6) involves a *function of* α, a real-valued parameter. However, as we will see, it turns out that $\text{Cost}_s^*(\alpha)$ and $\text{Cost}_s^\tau(\alpha)$ are simple functions of α that are easy to analytically manipulate.

Observe that, by definition (4.5), the optimal cost $\text{Cost}_s^*(\alpha)$ takes the form

$$\text{Cost}_s^*(\alpha) = \begin{cases} \text{Cost}_s^0(\alpha), & \alpha \in \mathcal{R}_s^0, \\ \vdots & \vdots \\ \text{Cost}_s^{s-1}(\alpha), & \alpha \in \mathcal{R}_s^{s-1}, \end{cases} \tag{4.7}$$

where $\mathcal{R}_s^\tau \equiv \left\{\alpha : \min_{0 \le \tau' < s} \text{Cost}_s^{\tau'}(\alpha) = \text{Cost}_s^\tau(\alpha)\right\}$; this is the set of values for the calcium at the sth timestep such that the most recent changepoint occurred at time τ. Furthermore, by inspection of (4.4), we see that $\text{Cost}_s^\tau(\alpha)$ is itself a quadratic function of α for all τ. Thus, $\text{Cost}_s^*(\alpha)$ is in fact *piecewise quadratic*. This means that in order to efficiently store the function $\text{Cost}_s^*(\alpha)$, we must simply keep track of the regions $\mathcal{R}_s^0, \ldots, \mathcal{R}_s^{s-1}$, as well as the three coefficients (constant, linear, quadratic) that define the quadratic function corresponding to each region. We will now present a small toy example illustrating how the recursion (4.6) can be used to build up optimal cost functions, each of which is piecewise quadratic.

Example 4.1 *Consider the simple dataset* $y = [1.00, 0.98, 0.96, \ldots]$ *with* $\lambda = \frac{1}{2}$ *and* $\gamma = 0.98$*.*

We start with $\text{Cost}_1^*(\alpha)$*, which is just a quadratic centered around* y_1*,*

$$\text{Cost}_1^*(\alpha) = \text{Cost}_1^0(\alpha) = \frac{1}{2}(y_1 - \alpha)^2 = \frac{1}{2}(1.00 - \alpha)^2, \quad \alpha \in \mathcal{R}_1^0 \equiv [0, \infty).$$

Then, at the next time point, we form $\text{Cost}_2^*(\alpha)$ *based on* (4.6)*,*

$$\begin{aligned}
\text{Cost}_2^*(\alpha) &= \min\left\{\text{Cost}_1^*(\alpha/\gamma), \min_{\alpha'} \text{Cost}_1^*(\alpha') + \lambda\right\} + \frac{1}{2}(y_2 - \alpha)^2 \\
&= \min\left\{\frac{1}{2}(1 - \alpha/\gamma)^2, 0 + \frac{1}{2}\right\} + \frac{1}{2}(0.98 - \alpha)^2 \\
&= \begin{cases} \frac{1}{2}(1 - \alpha/\gamma)^2 + \frac{1}{2}(0.98 - \alpha)^2, & \alpha \in \mathcal{R}_2^0 \equiv [0, 2\gamma) \\ \frac{1}{2} + \frac{1}{2}(0.98 - \alpha)^2, & \alpha \in \mathcal{R}_2^1 \equiv [2\gamma, \infty) \end{cases}.
\end{aligned}$$

Again, using the recursion (4.6) *we obtain the next optimal cost function,*

$$\begin{aligned}
\text{Cost}_3^*(\alpha) &= \min\left\{\text{Cost}_2^*(\alpha/\gamma), \min_{\alpha'} \text{Cost}_2^*(\alpha') + \lambda\right\} + \frac{1}{2}(y_3 - \alpha)^2 \\
&= \min\left\{\text{Cost}_2^*(\alpha/\gamma), \frac{1}{2}\right\} + \frac{1}{2}(0.96 - \alpha)^2 \\
&= \begin{cases} \frac{1}{2} + \frac{1}{2}(0.96 - \alpha)^2, & \alpha \in \mathcal{R}_3^2 \equiv \gamma^2\left\{\left[0, 1 - \frac{1}{\sqrt{1+\gamma^2}}\right) \cup \left[1 + \frac{1}{\sqrt{1+\gamma^2}}, \infty\right)\right\} \\ \frac{1}{2}(1 - \alpha/\gamma^2)^2 + \frac{1}{2}(0.98 - \alpha/\gamma)^2 + \frac{1}{2}(0.96 - \alpha)^2, & \alpha \in \mathcal{R}_3^0 \equiv \gamma^2\left[1 - \frac{1}{\sqrt{1+\gamma^2}}, 1 + \frac{1}{\sqrt{1+\gamma^2}}\right) \end{cases}.
\end{aligned}$$

We note that $\text{Cost}_3^*(\alpha)$ *is defined over just* $\mathcal{R}_3^0$ *and* $\mathcal{R}_3^2$*. This example is displayed in Figure 4.2.*

Although we have shown how to efficiently build optimal cost functions $\text{Cost}_s^*(\alpha)$ from $s = 1, \ldots, T$, it remains to establish that these cost functions can be used to determine the optimal changepoints, that is, the values of $\tau_1, \ldots, \tau_k$ that solve (3.4). These can be obtained by finding the value of τ that satisfies

$$\tau^*(s) = \{\tau : \min_{\alpha} \text{Cost}_s^\tau(\alpha) = \min_{\alpha} \text{Cost}_s^*(\alpha)\} \tag{4.8}$$

for $\tau^*(T), \tau^*(\tau^*(T)), \ldots$ until 0 is obtained. Full details are provided in Algorithm 4.1. To summarize, we have developed a recursive algorithm for solving (3.4) using the recursions in Proposition 4.1.

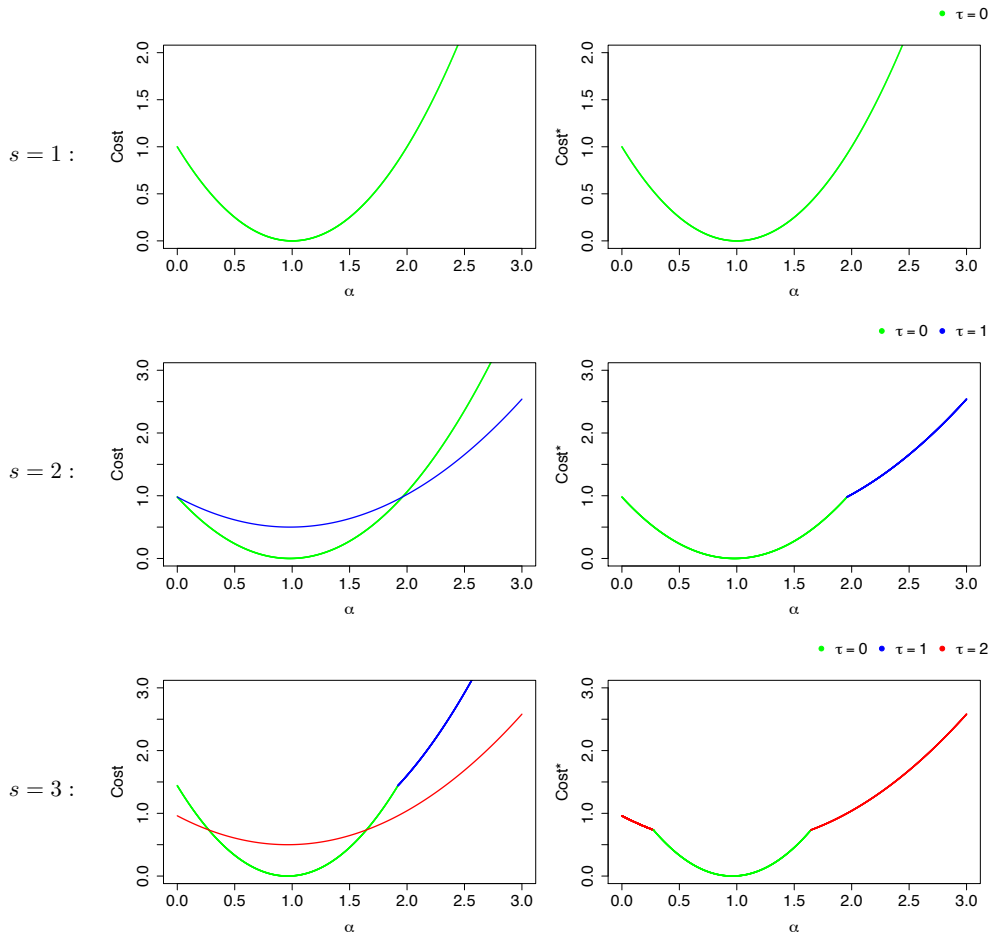


Figure 4.2: Evolution of Cost_s^τ and $\text{Cost}_s^*(\alpha)$ for Example 4.1. The left-hand panels display the functions $\text{Cost}_{s-1}^*(\alpha/\gamma) + \frac{1}{2}(y_s - \alpha)^2$ and $\min_{\alpha'} \text{Cost}_{s-1}^*(\alpha') + \lambda + \frac{1}{2}(y_s - \alpha)^2$, and the right-hand panels show the function $\text{Cost}_s^*(\alpha)$, which is the minimum of those two functions. Rows index the timesteps, $s = 1, 2, 3$. The functions are colored based on the timestep of the most recent changepoint, that is, the value of τ corresponding to $\mathcal{R}_s^\tau$. *Top:* When $s = 1$, $\text{Cost}_1^*(\alpha) = \frac{1}{2}(y_1 - \alpha)^2$; this corresponds to the region $\mathcal{R}_1^0 = [0, \infty)$. *Center:* When $s = 2$, $\text{Cost}_2^*(\alpha)$ is the minimum of two quantities: $\text{Cost}_1^*(\alpha/\gamma) + \frac{1}{2}(y_2 - \alpha)^2$, which corresponds to the most recent changepoint being at timestep zero, and $\min_{\alpha'} \text{Cost}_1^*(\alpha') + \lambda + \frac{1}{2}(y_2 - \alpha)^2$, which corresponds to the most recent changepoint being at timestep one. These two functions are shown on the left-hand side, and $\text{Cost}_2^*(\alpha)$ is shown on the right-hand side. *Bottom:* When $s = 3$, $\text{Cost}_3^*(\alpha)$ is calculated similarly; see Example 4.1 for additional details.

Algorithm 4.1: A functional pruning algorithm for solving (3.4)

Initialize: Compute $\text{Cost}_1^*(\alpha) := \text{Cost}_1^0(\alpha) = \frac{1}{2}(y_1 - \alpha)^2$, and set $\mathcal{R}_1^0 = [0, \infty)$

foreach *timestep* $s = 2, \ldots, T$ **do**

  Calculate and store

  $\text{Cost}_s^*(\alpha) := \min\{\text{Cost}_{s-1}^*(\alpha/\gamma), \min_{\alpha'} \text{Cost}_{s-1}^*(\alpha') + \lambda\} + \frac{1}{2}(y_s - \alpha)^2$

  Set $\mathcal{R}_s^{s-1} = \{\alpha : \text{Cost}_s^*(\alpha) = \min_{\alpha'} \text{Cost}_{s-1}^*(\alpha') + \lambda + \frac{1}{2}(y_s - \alpha)^2\}$

  foreach $\tau = 0, \ldots, s-1$ **do**

    $\mathcal{R}_s^\tau = (\gamma \mathcal{R}_{s-1}^\tau) \cap (\mathcal{R}_s^{s-1})^c$

  end

end

Initialize list of changepoints $cp := (T)$

Set the current changepoint $\tau^{cur} := T$

Initialize list of estimated calcium concentrations $c := ()$

while $\tau^{cur} > 0$ **do**

  $\tau^{prev} := \tau^{cur}$

  Determine the most recent changepoint

  $$\tau^{cur} := \left\{ \tau : \underset{\alpha}{\text{argmin}} \{\text{Cost}_{\tau^{prev}}^*(\alpha)\} \in \mathcal{R}_{\tau^{prev}}^\tau \right\}$$

  Determine the calcium concentration at τ^{prev}, $\alpha^* := \underset{\alpha \in \mathcal{R}_{\tau^{prev}}^{\tau^{cur}}}{\text{argmin}} \{\text{Cost}_{\tau^{prev}}^*(\alpha)\}$

  Update list of changepoints $cp := (\tau^{cur}, cp)$

  Update list of calcium concentrations, $c := (\alpha^*, c)$

  foreach *timestep* $s = (\tau^{prev} - 1), \ldots, (\tau^{cur} + 1)$ **do**

    Calculate calcium concentration, α^*/γ, and then append to list, $c := (\alpha^*/\gamma, c)$

    Scale $\alpha^* := \alpha^*/\gamma$

  end

end

Output : Set of changepoints cp, number of changepoints $k := \text{card}(cp)$, and estimated calcium concentrations c.

Example 4.2 *"Example 1 revisited"*

We return to Example 4.1 to illustrate how (4.8) *can be used to determine the optimal changepoints. In the interest of simplicity, we assume that* $T = 3$*; in other words, we have observed all of the data. Then,* $\tau^*(3) = \{\tau : \min_\alpha \text{Cost}_3^\tau(\alpha) = \min_\alpha \text{Cost}_3^*(\alpha)\}$*, where*

$$\min_\alpha \text{Cost}_3^\tau(\alpha) = \begin{cases} \min_\alpha \text{Cost}_3^2(\alpha) = 0.73, & \alpha \in \mathcal{R}_3^2 \\ \min_\alpha \text{Cost}_3^0(\alpha) = 5.4 \times 10^{-8}, & \alpha \in \mathcal{R}_3^0 \end{cases}.$$

Therefore, the most recent changepoint is $\tau^*(3) = 0$*. In fact, since the most recent changepoint is at timestep 0, we say that there are no changepoints.*

Algorithm 4.1 is an instance of the class of functional pruning algorithms proposed in Maidstone et al. (2017b).

Computational Time Of Functional Pruning

We saw in Example 4.1 that Proposition 4.1 can lead to a recursive algorithm for solving the problem of interest (3.4). At first glance, since $\text{Cost}_s^*(\alpha)$ is piecewise quadratic with s regions (4.7), and our recursive algorithm requires computing $\text{Cost}_1^*(\alpha), \ldots, \text{Cost}_T^*(\alpha)$, it appears that a total of $1 + 2 + \ldots + T = O(T^2)$ operations must be performed in order to deconvolve a fluorescence trace of length T. Critically, however, this is not the case. This is because, in practice, $\text{Cost}_s^*(\alpha)$ is piecewise quadratic with *substantially fewer than* s *regions*, as we saw in Figure 4.1. To see this, recall from (4.7) that the τth region up to timestep s is defined as $\mathcal{R}_s^\tau \equiv \left\{\alpha : \min_{0 \le \tau' < s} \text{Cost}_s^{\tau'}(\alpha) = \text{Cost}_s^\tau(\alpha)\right\}$. However, if $\mathcal{R}_s^\tau$ is the empty set — that is, if there is no α such that $\min_{0 \le \tau' < s} \text{Cost}_s^{\tau'}(\alpha) = \text{Cost}_s^\tau(\alpha)$ — then $\text{Cost}_s^*(\alpha)$ is, in fact, not a function of the τth region.

In practice, $\mathcal{R}_s^\tau$ will often be the empty set. For instance, see Figure 4.1. We note that in this example, at timestep $s = 40$, the optimal cost function is only a function of three

regions,

$$\text{Cost}^*_{40}(\alpha) = \begin{cases} 1.88\alpha^2 - 0.17\alpha + 2.08, & \alpha \in \mathcal{R}^{37}_{40} \equiv [0, 0.06] \\ 142.08\alpha^2 - 39.60\alpha + 3.85, & \alpha \in \mathcal{R}^{20}_{40} \equiv [0.06, 0.22] \\ 0.50\alpha^2 - 0.10\alpha + 2.10, & \alpha \in \mathcal{R}^{39}_{40} \equiv [0.22, \infty) \end{cases} \cdot$$

In a similar way, in Example 4.1, we saw that $\text{Cost}^*_3(\alpha)$ was a function of two regions.

Therefore, though its worst-case performance is upper-bounded by $O(T^2)$, in practice, Algorithm 4.1 is typically *much* faster than this. In Appendix B.6 we show that the maximum number of regions, $\max_{s=0,\ldots,T} |\{j : \mathcal{R}^j_s \neq \emptyset, 0 \leq j \leq s-1\}|$, is a small fraction of T; for $T = 100,000$, fewer than 30 regions are required.

Furthermore, by slightly modifying Theorem 6.1 of Maidstone et al. (2017b), we can show that Algorithm 4.1 is no worse than the algorithm proposed in Chapter 3. In fact, as shown in Figure 4.3, Algorithm 4.1 is typically up to a thousand times faster than that of Chapter 3 on a fluorescence trace of length 100,000. In simulations, our `C++` implementation of Algorithm 4.1 runs in less than one second on traces of length 100,000.

4.2.3 An Efficient Algorithm To Solve The Constrained Problem (3.3)

As stated in the introduction, our main interest is to solve (3.3) for the global optimum. Problem (3.3) differs from problem (3.4) in that there is an additional constraint that enforces biological reality: firing neurons can only cause an increase, but not a decrease, in the calcium concentration. The algorithm in Chapter 3 cannot be used to solve (3.3), because it relies on the recursion in (4.2), which does not allow for any dependence in the calcium concentration before and after a changepoint. Thus, at the time of this writing, there are no algorithms available to efficiently solve (3.3) for the global optimum.

In this section we utilize a simple modification, due to Hocking et al. (2017), to the functional recursion (4.6) that ensures that the constraint $c_t - \gamma c_{t-1} \geq 0$ is satisfied. First, recall from (4.6) that $\text{Cost}^*_s(\alpha) = \min\left\{\text{Cost}^*_{s-1}(\alpha/\gamma), \min_{\alpha'} \text{Cost}^*_{s-1}(\alpha') + \lambda\right\} + \frac{1}{2}(y_s - \alpha)^2$, where we take the minimum over two terms, which result from adding an additional point y_s

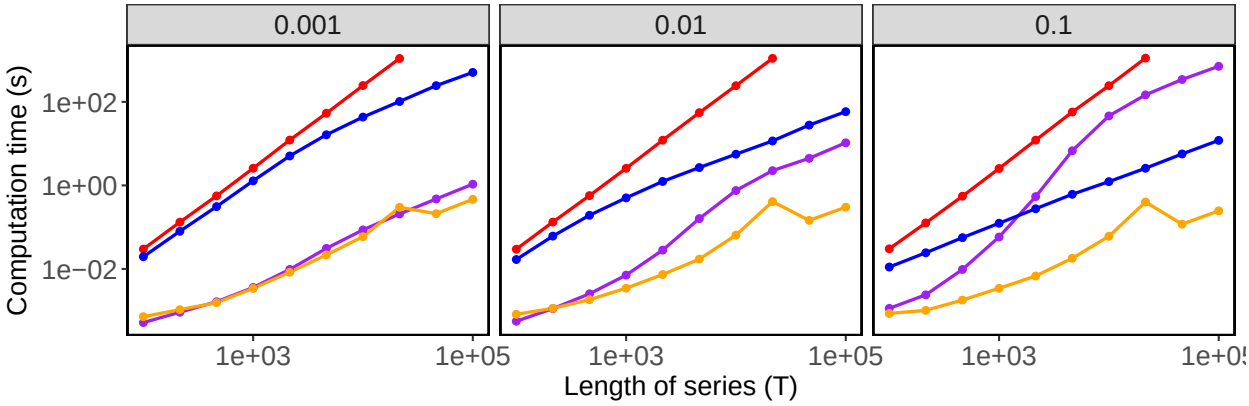


Figure 4.3: Timing comparisons between three algorithms for solving (3.3) and (3.4) with $\lambda = 1$. Functional pruning approach used in Algorithm 4.1 (orange) and Algorithm B.1 (purple), and two algorithms from Chapter 3: one based on recursion (4.2) (red), and one based on an improvement to (4.2) called inequality pruning that makes use of ideas from Killick et al. (2012) (blue). Fifty sample datasets are simulated according to (2.1) with coefficient $\beta_0 = 0$, $\beta_1 = 1$, decay parameter $\gamma = 0.998$, normal errors $\epsilon_t \overset{\text{ind}}{\sim} \text{N}(0, \sigma = 0.15)$, Poisson distributed spikes $z_t \overset{\text{ind}}{\sim} \text{Pois}(\theta)$ where $\theta \in \{0.1, 0.01, 0.001\}$, and initial calcium value $c_1 \sim \text{Pois}(\theta)$. Standard errors are on average $< 0.1\%$ of the average computation time. Panels correspond to different values of θ. Timing results were obtained on an Intel Xeon E5-2620 2.0 GHz processor.

to the current segment, $\text{Cost}^*_{s-1}(\alpha/\gamma) + \frac{1}{2}(y_s - \alpha)^2$, and adding a new candidate changepoint at $s-1$ and starting a new segment at the sth timestep, $\min_{\alpha'} \text{Cost}^*_{s-1}(\alpha') + \lambda + \frac{1}{2}(y_s - \alpha)^2$.

In the latter case, if there is a spike at the sth timestep, then in order to enforce the positivity constraint, $z_s = c_s - \gamma c_{s-1} \geq 0$, the term $\min_{\alpha'} \text{Cost}^*_{s-1}(\alpha') + \lambda$ in (4.6) needs to be modified to $\min_{\alpha':\alpha \geq \alpha'} \text{Cost}^*_{s-1}(\alpha'/\gamma) + \lambda$. Therefore, we replace (4.6) with

$$\text{Cost}^*_s(\alpha) = \min\left\{\text{Cost}^*_{s-1}(\alpha/\gamma), \min_{\alpha':\alpha \geq \alpha'} \text{Cost}^*_{s-1}(\alpha'/\gamma) + \lambda\right\} + \frac{1}{2}(y_s - \alpha)^2, \qquad (4.9)$$

and we replace (4.4) with

$$\text{Cost}_s^\tau(\alpha) \equiv \min_{\alpha':\alpha' \leq \gamma^{\tau-s}\alpha} \left[\text{Cost}_\tau^*(\alpha') + \frac{1}{2} \sum_{t=\tau+1}^{s} \left(y_t - \alpha\gamma^{t-s}\right)^2 + \lambda \right]. \tag{4.10}$$

We note that this is a slight abuse of notation since $\text{Cost}_s^*(\alpha)$ and $\text{Cost}_s^\tau(\alpha)$ take on different definitions depending on the optimization problem ((3.3) or (3.4)). Equations (4.9) and (4.10) can be used to develop an efficient recursive algorithm to solve problem (3.3). Details of the algorithm itself are included in Appendix B.2. A continuation of Example 4.1 that solves (3.3) is included in Appendix B.5. Figure 4.3 shows the running time of solving (3.3).

4.2.4 Solving (2.1) *for non-zero intercept* β_0

Thus far, we have considered (2.1) with $\beta_0 = 0$. To accommodate the possibility of nonzero baseline calcium, we consider the problem

$$\underset{c_1,\ldots,c_T,z_2,\ldots,z_T,\beta_0}{\text{minimize}} \left\{ \frac{1}{2} \sum_{t=1}^{T} \left(y_t - (\beta_0 + c_t)\right)^2 + \lambda \sum_{t=2}^{T} 1_{(z_t \neq 0)} \right\} \text{ subject to } z_t \geq c_t - \gamma c_{t-1}. \tag{4.11}$$

Instead of directly solving (4.11) with respect to $(c_1, \ldots, c_T, z_2, \ldots, z_T, \beta_0)$, we consider a fine grid of values for β_0, and solve (3.3) with $y - \beta_0$ using Algorithm B.1, for each value of β_0 considered. The solution to (4.11) is the set $\{\hat{c}_1, \ldots, \hat{c}_T, \hat{z}_2, \ldots, \hat{z}_T, \beta_0\}$ corresponding to the value of β_0 that led to the the smallest value of the objective, over all values of β_0 considered.

4.2.5 Solving (2.1) *with additional spike constraints*

The methods used to solve (3.3) and (3.4) can also be used to solve the related nonconvex problem

$$\underset{c_1,\ldots,c_T}{\text{minimize}} \left\{ \frac{1}{2} \sum_{t=1}^{T} (y_t - c_t)^2 \right\} \text{ subject to } c_t - \gamma c_{t-1} \geq z_{\min} \text{ or } c_t - \gamma c_{t-1} = 0, \tag{4.12}$$

proposed in Friedrich et al. (2017). In Appendix B.4, we examine this proposal more closely. Remarkably, we show that Algorithm B.1 can be generalized to solve

$$\underset{c_1,\ldots,c_T}{\text{minimize}} \left\{ \frac{1}{2}\sum_{t=1}^{T}(y_t - c_t)^2 + \lambda \sum_{t=2}^{T} 1_{\{c_t - \gamma c_{t-1} \neq 0\}} \right\} \text{ subject to } c_t - \gamma c_{t-1} \geq z_{\min} \text{ or } c_t - \gamma c_{t-1} = 0 \tag{4.13}$$

exactly! We note that this is equivalent to (4.12) by taking $\lambda = 0$.

4.3 Real Data Experiments

In this section, we illustrate the performance of the solution to (3.3) for spike deconvolution across a number of datasets, which were aggregated as part of the recent `spikefinder` challenge (`http://spikefinder.codeneuro.org/`). Each dataset consists of both calcium and electrophysiological recordings for a single cell. As part of the `spikefinder` challenge, all data recordings were standardized by resampling to 100Hz and linear trends were removed from the calcium trace via preprocessing steps described in Theis et al. (2016).

Throughout this section, due to computation considerations, the solutions to (3.3) and (3.4) are obtained using slight modifications of Algorithm 4.1 and Algorithm B.1. These modifications are described in Appendix B.3. Additionally, since the `spikefinder` data removed linear trends from the raw calcium trace, we do not estimate β_0 in (2.1). Instead, we set $\beta_0 = 0$; our empirical results suggest that estimation of β_0 may not be necessary.

In our experiments, we will treat the spikes ascertained using electrophysiological recording as the "ground truth", and will quantify the ability of spike deconvolution algorithms to recover these ground truth spikes on the basis of the calcium recordings. The data sets differ in terms of the choice of calcium indicator (GCaMP5, GCaMP6, jRCAMP, jRGECO, OGB), scanning technology (AOD, galvo, and resonant), and circuit under investigation (V1 and retina).

Throughout this section, we compare our proposal (3.3) to its convex relaxation (3.2)

replicated here for convenience,

$$\underset{c_1,\ldots,c_T,z_2,\ldots,z_T}{\text{minimize}} \left\{ \frac{1}{2}\sum_{t=1}^{T}\left(y_t - c_t\right)^2 + \lambda|c_1| + \lambda\sum_{t=2}^{T}|z_t| \right\} \text{ subject to } z_t = c_t - \gamma c_{t-1} \geq 0,$$

proposed by Friedrich and Paninski (2016) and Friedrich et al. (2017). Friedrich et al. (2017) developed a very fast algorithm to solve (3.2); in simulated examples their algorithm solves (3.2) approximately 40-60× faster than Algorithm 4.1 and 40-900× faster than Algorithm B.1. This is not surprising, since (3.2) is a convex problem whereas (3.3) and (3.4) are nonconvex problems. Moreover, in practical applications, Algorithm 4.1 and Algorithm B.1 are often fast enough. Indeed, de Vries et al. (2020) uses Algorithm B.1 to deconvolve traces from nearly 60,000 neurons.

Since the solution to (3.2) often results in many small non-zero elements of $\hat{z}_t$, we consider post-thresholding. That is, given $\hat{z}_2, \ldots, \hat{z}_T$ that solve (3.2), and a threshold $L > 0$, we set $\tilde{z}_t = \hat{z}_t 1_{(\hat{z}_t > L)}$; in other words, we conclude that a spike is present only if $\hat{z}_t > L$.

In Section 4.3.1 we compare our proposed approach (3.3) to (3.2) on data from the `spikefinder` challenge. We describe our experimental approach in Section 4.3.1. Section 4.3.1 illustrates these methods for a single cell, and in Section 4.3.1, we examine results for all datasets considered in the `spikefinder` challenge. In Section 4.3.2, we illustrate on a real-data example that solving (3.3) gives superior estimates than solving (3.4). In Section 4.3.3, we compare the estimated increase in calcium due to a spike (using (3.3)) to the actual number of recorded spikes (based on the ground truth electrophysiological recordings).

`R` code to reproduce all experiments is available on GitHub at `https://github.com/jewellsean/fast-nonconvex-experiments`.

4.3.1 Comparison Of (3.3) *To* (3.2) *On Data From The* `spikefinder` *Challenge*

Description Of Methods

We now describe the methods that will be used in the next two sections. Our main objective is to accurately estimate the times at which spikes occur. Thus, we use two measures

that directly compare two spike trains, both of which have been used extensively in the neuroscience literature (Quiroga and Panzeri, 2009; Reinagel and Reid, 2000; Gerstner et al., 2014): (i) van Rossum distance with timescale parameter $\tau = 0.1$ (van Rossum, 2001; Houghton and Kreuz, 2012); and (ii) Victor-Purpura distance with cost parameter 10 (Victor and Purpura, 1997, 1996). We also use an additional measure: (iii) the correlation between two downsampled spike trains; details of this measure are provided in Theis et al. (2016). As we will see, measures (i) and (ii) are sensitive to the timing of spikes, whereas measure (iii) is somewhat insensitive to the timing of the spikes, and instead quantifies the similarity between the spike rates.

To analyze the performances of the proposals (3.3) and (3.2) over a single fluorescence trace, we take a training/test set approach. Given a fluorescence trace of length T, the first $\lfloor T/2 \rfloor$ timesteps are used in the training set, and the remainder are used for the test set. We solve (3.3) and (3.2) for a range of values of the tuning parameter λ on the training set; in the case of (3.2) we also use a range of threshold values L.

For all tuning parameter values considered, we apply the three measures mentioned earlier to the estimated and true spike trains, and select the tuning parameter values that optimize these measures on the training set. We then apply (3.3) and (3.2) to the test set with the selected values of the tuning parameters, and evaluate test set performance.

As pointed out by Pachitariu et al. (2017), estimating the decay rate γ in (2.1) is difficult. Therefore, as in Pachitariu et al. (2017), we categorize calcium indicators into three groups based on their decay properties. As in Vogelstein et al. (2010), within each calcium indicator rate category, we set $\gamma = 1 - \frac{\Delta}{\phi}$, where Δ is 1 / (frame rate), and ϕ is a time-scale parameter based on the category, defined as

$$\phi = \begin{cases} 0.7, & \text{fast category} \\ 1.25, & \text{medium category} \\ 2, & \text{slow category} \end{cases} \cdot$$

For example, in Figure 4.4, GCaMP6f is classified as a fast indicator and the data is recorded

at 100Hz. Therefore, we take $\gamma \approx 0.986$.

In practice, users typically do not have the benefit of a training set to select the tuning parameter value λ to solve (3.3) or (3.4). Therefore, we recommend using the procedure of Friedrich et al. (2017) and Pnevmatikakis et al. (2016), summarized in Appendix A.3, or the procedure proposed in de Vries et al. (2020), which selects λ based on the firing rate, decay rate γ, and estimated signal-to-noise ratio.

Results For A Single Cell

In Figure 4.4, we illustrate this procedure for cell 13, GCaMP6f, V1, from Chen et al. (2013). Each row corresponds to one of the measures described in Section 4.3.1. The left column displays these measures on the training set, for the solution to (3.3) with different values of λ, and for the post-thresholded solution to (3.2) with different values of λ and L. The right column shows the fluorescence trace along with the estimated spikes, on the test set, using tuning parameters selected on the training set.

There are a number of important observations to draw from Figure 4.4. As measured by van Rossum and Victor-Purpura, the estimated spikes from (3.3) are much more accurate than those estimated (and post-thresholded) using the convex relaxation (3.2). This agrees with our visual inspection of the right hand panel: the estimated spikes from problem (3.3) more closely match the number and timings of the true spikes than those estimated from problem (3.2).

By contrast, if performance is measured by correlation, then the estimated spikes obtained from (3.2) result in slightly better performance than the estimated spikes from (3.3). However, in the training set there are 75 true spikes, whereas (3.2) outperforms (3.3) when approximately 200 spikes are estimated. Therefore, selecting the tuning parameter for (3.2) based on correlation leads to a *substantial overestimate* of the number of spikes, and therefore poor overall accuracy in the number and timing of the spikes. This pattern has been observed in other ℓ_1 regularization problems (Zou, 2006; Maidstone et al., 2017a), and persists across cells in the `spikefinder` data (results not shown).

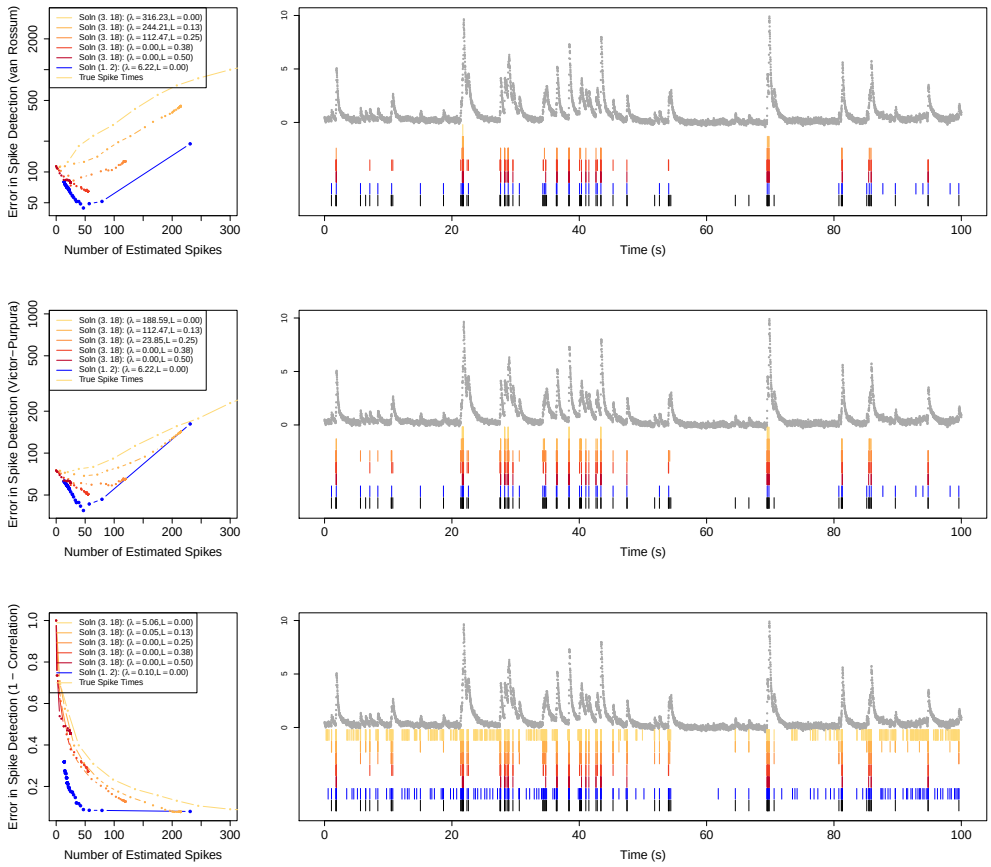


Figure 4.4: Illustrative example for cell 13, GCaMP6f, V1, from Chen et al. (2013) after preprocessing; see Theis et al. (2016). Different spike measures are displayed in each row. *Left:* Performances of the post-thresholded solution to (3.2) and the solution to (3.3). *Right:* The cell's fluorescence trace is displayed in grey. The estimated spikes on the test set from the "best" choice of the tuning parameter λ, as determined by either van Rossum, Victor-Purpura, or a correlation-based measure on the training set, are displayed under the fluorescence trace. The true spike times, as determined using electrophysiological recording, are shown in black. The colors in the left-hand panels correspond to the colors in the right-hand panel.

To summarize, when van Rossum and Victor-Purpura distance are used to evaluate performance, our proposal (3.3) substantially outperforms the approach in (3.2). When performance is evaluated using correlation, the performance of (3.2) is slightly better than that of (3.3); however, this better performance is achieved when far too many spikes are estimated, indicating that correlation is a poor choice for quantifying the accuracy of spike detection.

Results For All Datasets In The `spikefinder` *Challenge*

In this section, we examine the performance of the solutions to (3.3) and (3.2) on all datasets collected as part of the `spikefinder` challenge. For the ten datasets included in this challenge, Table B.1 tabulates the calcium indicator; circuit; publishing authors; average, minimum, and maximum fluorescence trace length; the number of cells measured; and the timescale classification. In total, there are 174 traces, each of which contains fewer than 100,000 timesteps. We analyze these 174 cells as described in Section 4.3.1.

Figure 4.5 compares the test set performance, with respect to the van Rossum, Victor-Purpura, and correlation measures, for each of the 174 cells. As measured by the van Rossum and Victor-Purpura distance, the solution to (3.3) outperforms the solution to (3.2). However, under the correlation measure, the solution to (3.2) achieves higher correlations than the solution to (3.3). These results are consistent with those on a single cell where it was shown that van Rossum and Victor-Purpura accurately estimate spike times, whereas correlation yields a cruder measure of spike rate and encourages overestimation of the number of spikes.

4.3.2 The Solution To (3.3) *Outperforms The Solution To* (3.4)

As mentioned earlier, in this chapter we have developed not only an algorithm for solving (3.4) that is much faster than the algorithm proposed in Chapter 3, but also an algorithm for solving (3.3), which cannot be solved using techniques from Chapter 3. By incorporating the fact that a firing neuron causes an increase, but never a decrease, in the calcium concentration, the estimated spikes from problem (3.3) are closer to the ground truth spikes

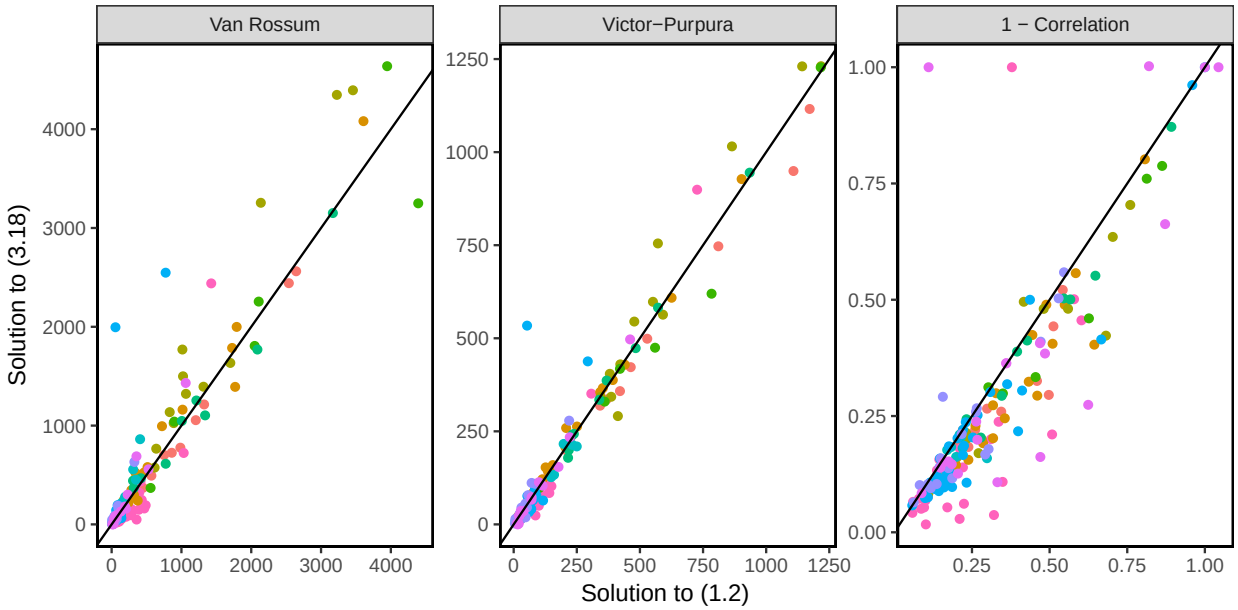


Figure 4.5: Optimal van Rossum, Victor-Purpura, and correlation measures for our proposal, (3.3), and a competing proposal, (3.2). Small values of the van Rossum and Victor-Purpura measures suggest accurate estimation of the timing and number of spikes, whereas a large value of the correlation measure suggests accurate estimation of the spike rate, though perhaps an overestimate of the number of spikes. Each dot represents the performance of (3.3) and (3.2) on a single cell, for each of the 174 cells. Cells are colored based on the dataset from which they were obtained (see Table B.1).

than the estimated spikes from (3.4). In practice, the solutions to (3.3) and (3.4) are typically quite similar; however, the solution to (3.3) benefits from greater interpretability. See Appendix B.8 for an example.

4.3.3 Comparison Of The Estimated Spike Magnitudes From (3.3) *To The True Number Of Spikes*

The data from the `spikefinder` challenge was resampled to 100Hz before we downloaded it. At this sampling frequency, since one timestep is just 1/100th of a second, there are very few timesteps with more than one true spike. Nonetheless, for instances where there is more than one spike in a single timestep, we wish to ask the question: Do larger values of the estimated spike magnitudes, $\hat{c}_t - \gamma\hat{c}_{t-1}$, correspond to more true spikes (as measured by electrophysiology) in the tth timestep?

Figure 4.6 investigates whether there is a relationship between the estimated spike magnitude $\hat{c}_t - \gamma\hat{c}_{t-1}$ and the number of spikes measured by electrophysiology at the tth timestep. Because the estimated spike magnitude of $\hat{c}_t - \gamma\hat{c}_{t-1}$ is not directly comparable across cells, for each cell we transform the magnitudes into percentiles. We then compare the percentile of $\hat{c}_t - \gamma\hat{c}_{t-1}$ to the true number of spikes within a 0.1 second window of t. Figure 4.6 displays the percentiles and the number of spikes across all 174 traces on a test set; tuning parameters were chosen to optimize the van Rossum distance on a training set. The left panel displays a loess curve fit to all ten datasets, and the right panel shows the loess curves along with 95% confidence intervals for each dataset. As expected, a larger value of $\hat{c}_t - \gamma\hat{c}_{t-1}$ is associated with more spikes in the ground truth data.

4.4 Discussion

Determining the times at which a neuron fires from a calcium imaging dataset is a challenging and important problem. In this chapter, we build upon the nonconvex approach for spike deconvolution proposed in Chapter 3. Though Chapter 3 proposed a tractable algorithm for solving the nonconvex problem, it is prohibitively slow to run on large populations of neurons

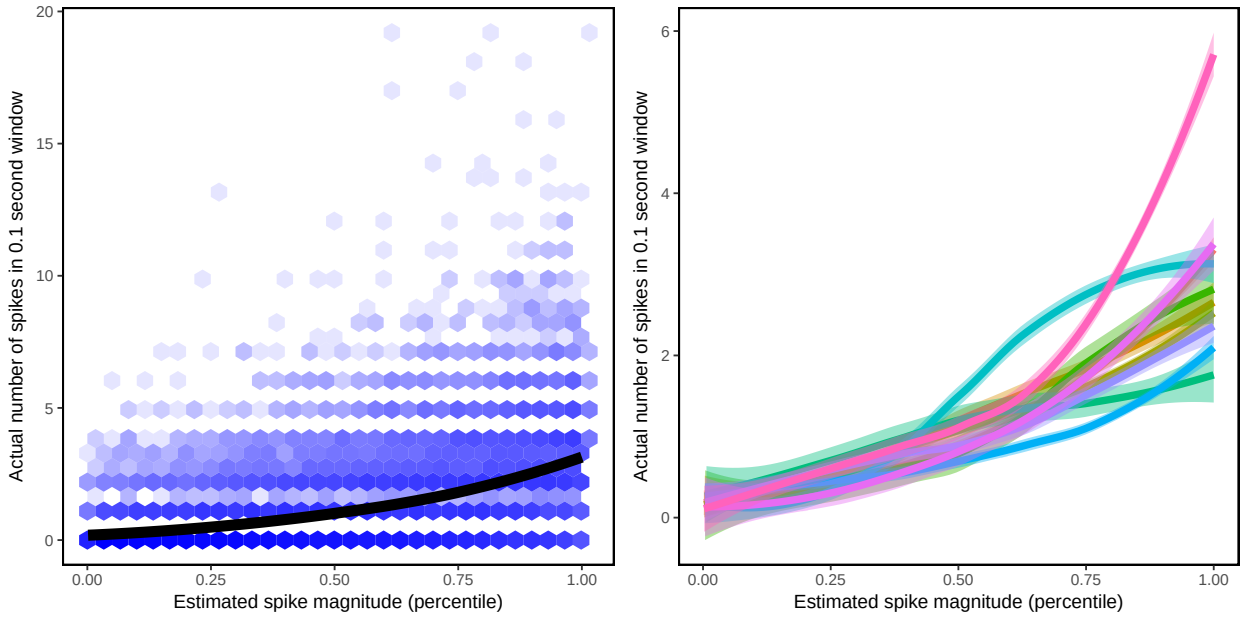


Figure 4.6: Large increases in the estimated spike magnitude, $\hat{c}_t - \gamma\hat{c}_{t-1}$, are associated with more true spikes, as measured by electrophysiology, at the tth timestep. For each cell, we transform the spike magnitudes into percentiles, and then compare the percentile of $\hat{c}_t - \gamma\hat{c}_{t-1}$ to the true number of spikes within a 0.1 second window of t. *Left:* For each cell in each of the ten datasets, we display each timestep for which a spike is estimated to occur; however, to avoid overplotting, hexagonal bins are used to represent points covered by the hexagon; darker colors indicate more points. The black curve represents the loess fit across all of the points. *Right:* Loess curves along with 95% confidence intervals for each dataset. Cells are colored based on the dataset from which they were obtained (see Table B.1). Details are provided in Section 4.3.3.

for which long recordings are available. The algorithm proposed in this chapter solves the optimization problem of Chapter 3 for fluorescence traces of 100,000 timesteps in less than a second. Moreover, Algorithm B.1 overcomes a limitation of Chapter 3 by avoiding "negative" spikes; that is, a decrease in the calcium concentration due to a spike. We show that these algorithms have excellent performance, relative to existing convex relaxations, as quantified by the van Rossum and Victor-Purpura measures, on datasets collected as part of the `spikefinder` challenge (`http://spikefinder.codeneuro.org/`). Moreover, Algorithm B.1 was recently used to decode data from nearly $60,000$ neurons in the Allen Institute for Brain Science's "platform paper" for the Allen Brain Observatory (de Vries et al., 2020).

In this chapter, we assume that the calcium concentration decays exponentially according to a first-order auto-regressive model. Although this is typically a good approximation, there are datasets for which—due to different experimental conditions—spike times estimated from (3.3) and (3.4) are systematically biased due to model misspecification. In future work, we propose to extend the functional pruning framework to more general calcium models.

In this chapter, we focus on developing *point estimates* of the times at which a neuron spikes. However, it is also of interest to propagate uncertainty from the deconvolution procedure to downstream analyses that rely on the spike times. We develop a framework for this task in Chapter 5.

Chapter 5

TESTING FOR A CHANGE IN MEAN AFTER CHANGEPOINT DETECTION

5.1 Introduction

Detecting structural changes in a time series is a fundamental problem in statistics, with a variety of applications (Bai and Perron, 1998, 2003; Muggeo and Adelfio, 2010; Schröder and Fryzlewicz, 2013; Futschik et al., 2014; Xiao et al., 2019; Harchaoui and Lévy-Leduc, 2007; Hotz et al., 2013). A structural change refers to the phenomenon that at certain (unknown) timepoints, the law of the data may change: that is, observations $y_1, \ldots, y_T$ are heterogeneous in the sense that $y_1, \ldots, y_\tau \sim F$, whereas $y_{\tau+1}, \ldots, y_T \sim G$, for distribution functions $F \neq G$. In the presence of possible structural changes, it is of interest not only to estimate the times at which these changes occur — that is, the value of τ — but also to conduct statistical inference on the estimated changepoints.

In this chapter, we consider a simple changepoint model

$$Y_t = \mu_t + \epsilon_t, \quad \epsilon_t \overset{\text{iid}}{\sim} \mathrm{N}(0, \sigma^2), \quad t = 1, \ldots, T, \tag{5.1}$$

and assume that $\mu_1, \ldots, \mu_T$ is piecewise constant, in the sense that $\mu_{\tau_j+1} = \mu_{\tau_j+2} = \ldots = \mu_{\tau_{j+1}}$, $\mu_{\tau_{j+1}} \neq \mu_{\tau_{j+1}+1}$, for $j = 0, \ldots, K-1$, where $0 = \tau_0 < \tau_1 < \ldots < \tau_K < \tau_{K+1} = T$, and where $\tau_1, \ldots, \tau_K$ represent the true changepoints. Changepoint detection refers to the task of estimating the changepoint locations $\tau_1, \ldots, \tau_K$, and possibly the number of changepoints K. A huge number of proposals for this task have been made in the literature, and can be roughly divided into two main classes. One class of proposals iteratively searches for one changepoint at a time (Vostrikova, 1981; Olshen et al., 2004; Fryzlewicz et al., 2014; Badagián et al., 2015; Anastasiou and Fryzlewicz, 2019); the canonical example of this approach is binary segmentation. Another class of proposals involves simultaneously estimating all changepoints

by solving a single optimization problem (Yao, 1988; Auger and Lawrence, 1989; Jackson et al., 2005; Tibshirani et al., 2005; Niu and Zhang, 2012; Killick et al., 2012; Haynes et al., 2017; Maidstone et al., 2017b; Jewell and Witten, 2018; Fearnhead et al., 2019; Hocking et al., 2018; Jewell et al., 2019b); examples include ℓ_0 segmentation and the fused lasso. We review these two classes in Section 5.2. Changepoint estimation and inference have also been studied from a Bayesian viewpoint (Fearnhead, 2006; Nam et al., 2012; Ruanaidh and Fitzgerald, 2012).

In the single changepoint setting, estimation and inference for the location of the changepoint have been studied in the asymptotic (Hinkley, 1970; Yao, 1987; Bai, 1994; James et al., 1987) and non-asymptotic (Enikeeva and Harchaoui, 2019) settings. These approaches are typically extended to the multiple changepoint setting by repeated application of tests for a single changepoint to sliding subsets of the data.

In the multiple changepoint setting, the multiscale approach of Frick et al. (2014) estimates the changepoints and provides confidence intervals for their locations and means. However, this approach aims to control the probability of falsely detecting a change, and can lose power in situations where there are many changes, particularly when they are hard to detect. Ma and Yau (2016) produce asymptotically valid confidence intervals under an asymptotic regime where all of the changepoints are detected with probability tending to one; this is unrealistic in many settings. To overcome these issues, Li et al. (2016) develop a multiscale procedure that controls the false discovery rate of detections, but they use a very weak definition of a "true changepoint"; in extreme cases, this could include an estimated changepoint that is almost as far as $T/2$ observations from an actual changepoint. Non-parametric approaches to estimate multiple changepoints, such as moving-sum or scan statistics, have also been proposed (Bauer and Hackl, 1980; Chu et al., 1995; Hušková, 1990), and their convergence rates have been studied (Eichinger et al., 2018).

Despite the extensive literature on changepoint estimation and inference, there is a gap between the procedures used by practitioners to estimate changepoints, and the statistical tools that are available to assess the uncertainty of these estimates. In particular, existing

approaches for changepoint inference suffer from several shortcomings:

1. Most classic results for testing the location of a single changepoint are asymptotic in nature, and result in complicated limiting distributions; furthermore, they cannot be directly extended to the multiple changepoint setting;

2. In the multiple changepoint setting, many of the theoretical results rely on specialized estimation procedures that are designed to facilitate inference. This means that there is a gap between the estimation procedures that are typically used in practice, and the theoretical results that are available;

3. Much of the work on changepoint inference focuses on providing confidence statements for the *location* of the changepoint. However, downstream analyses often also rely on the *size* of the shift in mean and not its precise location. Thus, available inference methods may be conducting inference on a quantity that is not of primary scientific interest.

As a result, this existing literature does not allow an applied scientist who wishes to estimate changepoints on a (finite sample) data set using a state-of-the-art estimation approach — such as binary segmentation or its variants, ℓ_0 segmentation, or the fused lasso — to quantify the uncertainty associated with those estimated changepoints, and in particular, to test for the presence of a change in mean.

To addresses these limitations, we consider testing the null hypothesis that there is no change in mean around an estimated changepoint, in the context of three very popular changepoint detection procedures: binary segmentation, ℓ_0 segmentation, and the fused lasso. Our interest lies not in determining whether there is a change at a precise location, but rather, whether there is a change in mean near an estimated changepoint. This is a challenging task since we must account for the estimation process when deriving the null distribution for a test statistic. A recent promising line of work was introduced by Hyun et al. (2016) and Hyun et al. (2018), who develop valid tests for changepoints estimated

with the generalized lasso and with binary segmentation, respectively. They leverage recent results for selective inference in the regression setting (Fithian et al., 2014, 2015; Tibshirani et al., 2016; Lee et al., 2016; Tian et al., 2018). However, a major disadvantage of their proposals is that, when defining p-values, they need to condition on much more information than is used to choose the null hypothesis that is tested. This is especially relevant since Fithian et al. (2014), Lee et al. (2016), and Liu et al. (2018) show that conditioning on extra information leads to a reduction in power.

In this chapter, our contributions are two-fold:

1. We implement selective inference for the change in mean after changepoint detection while conditioning on far less information than Hyun et al. (2016) and Hyun et al. (2018). This leads directly to a substantial increase in power;

2. We conduct inference not only on changepoints estimated via binary segmentation and fused lasso, as in Hyun et al. (2016) and Hyun et al. (2018), but also on changepoints estimated via ℓ_0 segmentation. This leads to a substantial improvement in empirical results, since ℓ_0 segmentation detects changepoints very accurately. We develop this framework in detail for the change-in-mean model, but the general ideas can be applied much more widely.

The rest of this chapter is organized as follows. In Section 5.2, we review the relevant literature on changepoint detection and inference. In Section 5.3, we introduce a framework for inference in changepoint detection problems, which allows us to test for a change in mean associated with a changepoint estimated on the same dataset. In Sections 5.4 and 5.5, we develop efficient algorithms that allow us to instantiate this framework in the special cases of binary segmentation (Vostrikova, 1981) and its variants (Olshen et al., 2004; Fryzlewicz et al., 2014), and ℓ_0 segmentation (Yao, 1987; Killick et al., 2012; Maidstone et al., 2017b); the case of the fused lasso (Tibshirani et al., 2016) is straightforward and addressed in Appendix C. Our framework is a substantial improvement over the existing approaches for

inference on the changepoints estimated using binary segmentation and its variants and the fused lasso; it is completely new in the case of ℓ_0 segmentation. After a preprint of this work appeared (Jewell et al., 2019a), another research group developed a related but less efficient dynamic programming approach to assess the uncertainty in changepoints estimated from ℓ_0 segmentation (Duy et al., 2020).

In Section 5.6, we examine the performance of our proposal, and compare it to some recent proposals from the literature, in a simulation study. In Section 5.7, we show that our procedure leads to additional discoveries versus existing methods on a dataset of chromosomal guanine-cytosine (G-C) content. Extensions are in Section 5.8, and some additional details are deferred to Appendix C.

Our new changepoint inference procedures are freely available in the `R` package `ChangepointInference`. Code and data to produce all figures are available at `https://jewellsean.github.io/changepoint-inference`.

5.2 Background

5.2.1 Changepoint Detection Algorithms

Binary Segmentation And Its Variants

The binary segmentation proposal of Vostrikova (1981) and its variants (Olshen et al., 2004; Fryzlewicz et al., 2014) search for changepoints by solving a sequence of local optimization problems. For the change-in-mean problem, these use the cumulative sum (CUSUM) statistic

$$g_{(s,\tau,e)}^{\top} y := \sqrt{\frac{1}{\frac{1}{|e-\tau|} + \frac{1}{|\tau+1-s|}}} (\bar{y}_{(\tau+1):e} - \bar{y}_{s:\tau}), \tag{5.2}$$

defined through a contrast $g_{(s,\tau,e)} \in \mathbb{R}^T$, which summarizes the evidence for a change at τ in the data $y_{s:e} := (y_s, \ldots, y_e)$ by the difference in the empirical mean of the data before and after τ (normalized to have the same variance for all τ).

In binary segmentation (Vostrikova, 1981), the set of estimated changepoints is simply the set of local CUSUM maximizers: the first estimated changepoint maximizes the CUSUM

statistic over all possible locations, $\hat{\tau}_1 = \underset{\tau \in [1:(T-1)]}{\text{argmax}} \left\{ |g_{(1,\tau,T)}^\top y| \right\}$. Subsequent changepoints are estimated at the location that maximizes the CUSUM statistic when we consider regions of the data between previously estimated changepoints. For example, the second estimated changepoint is $\hat{\tau}_2 = \underset{\tau \in [1:(T-1)] \setminus \hat{\tau}_1}{\text{argmax}} \left\{ |g_{(1,\tau,\hat{\tau}_1)}^\top y| 1_{(1 \leq \tau < \hat{\tau}_1)} + |g_{(\hat{\tau}_1,\tau,T)}^\top y| 1_{(\hat{\tau}_1 < \tau < T)} \right\}$. We continue in this manner until a stopping criterion is met. Variants of this procedure have been proposed to improve performance (Olshen et al., 2004; Fryzlewicz et al., 2014).

Simultaneous Estimation Of Changepoints

As an alternative to sequentially estimating changepoints, we can simultaneously estimate all changepoints by minimizing a penalized cost that trades off fit to the data against the number of changepoints (Yao, 1987; Killick et al., 2012; Maidstone et al., 2017b), i.e.

$$\underset{\substack{0=\tau_0<\tau_1<\cdots<\tau_K<\tau_{K+1}=T, \\ u_0,u_1,\ldots,u_K,K}}{\text{minimize}} \left\{ \frac{1}{2} \sum_{k=0}^{K} \sum_{t=\hat{\tau}_k+1}^{\hat{\tau}_{k+1}} (y_t - u_k)^2 + \lambda K \right\}. \tag{5.3}$$

This is equivalent to solving an ℓ_0 penalized regression problem

$$\underset{\mu \in \mathbb{R}^T}{\text{minimize}} \left\{ \frac{1}{2} \sum_{t=1}^{T} (y_t - \mu_t)^2 + \lambda \sum_{t=2}^{T} 1_{(\mu_t \neq \mu_{t-1})} \right\}, \tag{5.4}$$

in the sense that the vector $\hat{\mu}$ that solves (5.4) is piecewise constant with breakpoints at $\hat{\tau}_1, \ldots, \hat{\tau}_{\hat{K}}$, where $\hat{\tau}_1, \ldots, \hat{\tau}_{\hat{K}}$ are the changepoints that solve (5.3). Here, λ is a tuning parameter that specifies the improvement in fit to the data needed to add an additional changepoint.

Replacing the ℓ_0 penalty in (5.4) with an ℓ_1 penalty leads to the well-studied trend filtering or fused lasso optimization problem (Rudin et al., 1992; Tibshirani et al., 2005),

$$\underset{\mu \in \mathbb{R}^T}{\text{minimize}} \left\{ \frac{1}{2} \sum_{t=1}^{T} (y_t - \mu_t)^2 + \lambda \sum_{t=2}^{T} |\mu_t - \mu_{t-1}| \right\}. \tag{5.5}$$

5.2.2 Existing Methods For Inference On Changepoints Post-Detection

Suppose we wish to quantify the evidence for a set of estimated changepoints $\hat{\tau}_1, \ldots, \hat{\tau}_{\hat{K}}$. We could naively apply a standard z-test for the difference in mean around each estimated

changepoint. However, this procedure is fundamentally flawed since the same data is used to estimate the changepoints and thus to choose the hypothesis tests that we perform. Therefore, the z-statistic is not normally distributed under the null hypothesis. In the linear regression setting, Tibshirani et al. (2016) and Lee et al. (2016) have shown that it is possible to select and test hypotheses based on the same set of data, provided that we condition on the output of the hypothesis selection procedure.

Hyun et al. (2016) and Hyun et al. (2018) extend these ideas to the changepoint detection setting. For each changepoint $\hat{\tau}_j$ estimated using either binary segmentation, its variants, or the fused lasso, Hyun et al. (2018) propose to test whether there is a change in mean around $\hat{\tau}_j$. They construct the test statistic $\hat{d}_j \nu^\top Y$, where $\hat{d}_j$ is the sign of the estimated change in mean at $\hat{\tau}_j$, and ν is a T-vector of contrasts, defined as

$$\nu_t = \begin{cases} 0 & \text{if } t \leq \hat{\tau}_{j-1} \text{ or } t > \hat{\tau}_{j+1}, \\ \frac{1}{\hat{\tau}_j - \hat{\tau}_{j-1}} & \text{if } \hat{\tau}_{j-1} < t \leq \hat{\tau}_j, \\ -\frac{1}{\hat{\tau}_{j+1} - \hat{\tau}_j} & \text{if } \hat{\tau}_j < t \leq \hat{\tau}_{j+1}, \end{cases} \tag{5.6}$$

and consider the null hypothesis $H_0 : \hat{d}_j \nu^\top \mu = 0$ versus the one-sided alternative $H_1 : \hat{d}_j \nu^\top \mu > 0$. Since both $\hat{d}_j$ and ν are functions of the estimated changepoints themselves, it is clear that valid inference requires somehow conditioning on the estimation process in the spirit of Tibshirani et al. (2016) and Lee et al. (2016). Define $\mathcal{M}(y)$ to be the set of changepoints estimated from the data y, i.e., $\mathcal{M}(y) = \{\hat{\tau}_1, \ldots, \hat{\tau}_{\hat{K}}\}$. Then, it is tempting to define the p-value as

$$\Pr_{H_0}\left(\hat{d}_j \nu^\top Y \geq \hat{d}_j \nu^\top y \mid \mathcal{M}(Y) = \mathcal{M}(y)\right). \tag{5.7}$$

However, (5.7) is not immediately amenable to the selective inference framework proposed by Tibshirani et al. (2016) and Lee et al. (2016), which requires that the conditioning set be polyhedral; i.e., the conditioning set can be written as $\{y : \boldsymbol{A}y \leq b\}$ for a matrix $\boldsymbol{A}$ and vector b. Thus, in the case of binary segmentation, Hyun et al. (2018) condition on three additional quantities: (i) the order in which the estimated changepoints enter the

model, $\mathcal{O}(Y) = \mathcal{O}(y)$; (ii) the sign of the change in mean due to the estimated changepoints, $\Delta(Y) = \Delta(y) = \{\hat{d}_1, \ldots, \hat{d}_K\}$; (iii) $\Pi_\nu^\perp Y = \Pi_\nu^\perp y$, where $\Pi_\nu^\perp = I - \nu\nu^\top / ||\nu||_2^2$ is the orthogonal projection matrix onto the subspace that is orthogonal to ν. Conditions (i) and (ii) ensure that the conditioning set is polyhedral, whereas condition (iii) ensures that the test statistic is a pivot. This leads to the p-value

$$\Pr_{H_0}\left(\hat{d}_j \nu^\top Y \geq \hat{d}_j \nu^\top y \mid \mathcal{M}(Y) = \mathcal{M}(y), \mathcal{O}(Y) = \mathcal{O}(y), \Delta(Y) = \Delta(y), \Pi_\nu^\perp Y = \Pi_\nu^\perp y\right), \tag{5.8}$$

which can be easily computed because the conditional distribution of $\hat{d}_j \nu^\top Y$ is a Gaussian truncated to an interval, which is computationally tractable. For slightly different conditioning sets, Hyun et al. (2018) show similar results for variants of binary segmentation and for the fused lasso.

Importantly, Hyun et al. (2018) choose the conditioning set in (5.8) for computational reasons: there is no clear statistical motivation for conditioning on $\mathcal{O}(Y) = \mathcal{O}(y)$ and $\Delta(Y) = \Delta(y)$. Furthermore, it might be possible to account for the fact that changepoints are estimated from the data without conditioning on the full set $\mathcal{M}(Y) = \mathcal{M}(y)$. In fact, Fithian et al. (2014) argue that when conducting selective inference, it is better to condition on a larger set, since conditioning on more information reduces the Fisher information that remains in the conditional distribution of the data.

For this reason, in the regression setting, some recent proposals seek to increase the size of the conditioning set. Lee et al. (2016) propose to condition on just the selected model, rather than on the selected model and the corresponding coefficient signs, by considering all possible configurations of the signs of the estimated coefficients. Unfortunately, this comes at a significant computational cost. Continuing in this vein, Liu et al. (2018) partition the selected variables into high value and low value subsets, and then condition on the former and the variable of interest.

In this chapter, we develop new insights that allow us to test the null hypothesis that there is no change in mean at an estimated changepoint, without restriction to the polyhedral conditioning sets pursued by Hyun et al. (2018). This means that we do not need to use the

conditioning set in (5.8), and, in turn, leads to higher-powered tests. Additionally, since we do not need to condition on $\Delta(Y) = \Delta(y)$, we are able to consider two-sided tests of

$$H_0 : \nu^\top \mu = 0 \text{ versus } H_1 : \nu^\top \mu \neq 0, \tag{5.9}$$

rather than the one-sided tests considered by Hyun et al. (2018).

It is natural to ask whether we can avoid the complications of selective inference and use alternative approaches that control the false discovery rate (Benjamini and Hochberg, 1995; Benjamini et al., 2001; Barber et al., 2015; Candes et al., 2018). However, these alternatives are not suitable for the changepoint setting in the following sense. Often we do not want to know if a true changepoint is *exactly* at $\hat{\tau}_j$, but rather whether there is a true changepoint *near* $\hat{\tau}_j$; that is, we are willing to accept small estimation errors in the location of a changepoint. By suitable choice of ν in (5.9), we can test whether there is a change in mean near $\hat{\tau}_j$, where *near* can be defined appropriately for a given application. It is possible that application of, for example, knockoffs (Barber et al., 2015; Candes et al., 2018) would enable us to control the false discovery rate for the null hypotheses that the changes in mean are precisely at $\hat{\tau}_1, \ldots, \hat{\tau}_{\hat{K}}$. However, our experience with such methods is that they have almost no power to detect small to moderate changes in the mean, due to the large uncertainty in the precise location of the change.

5.2.3 Toy Example Illustrating The Cost Of Conditioning

In this section, we demonstrate that the power of a test of (5.9) critically depends on the size of the conditioning set. In Figure 5.1, we consider two choices for the conditioning set. In panel a), we condition on $\mathcal{M}(Y) = \mathcal{M}(y), \mathcal{O}(Y) = \mathcal{O}(y), \Delta(Y) = \Delta(y)$, and $\Pi_\nu^\perp Y = \Pi_\nu^\perp y$ (which is essentially the test proposed by Hyun et al. (2018)). In panel b) we condition on just $\mathcal{M}(Y) = \mathcal{M}(y)$ and $\Pi_\nu^\perp Y = \Pi_\nu^\perp y$. Observed data (grey points) are simulated according to (5.1) with the true underlying mean displayed in blue. 19-step binary segmentation is used to estimate changepoints, which are displayed as vertical lines, and are colored based on whether the associated p-value is less than 0.05 (blue) or greater than 0.05 (red). In this

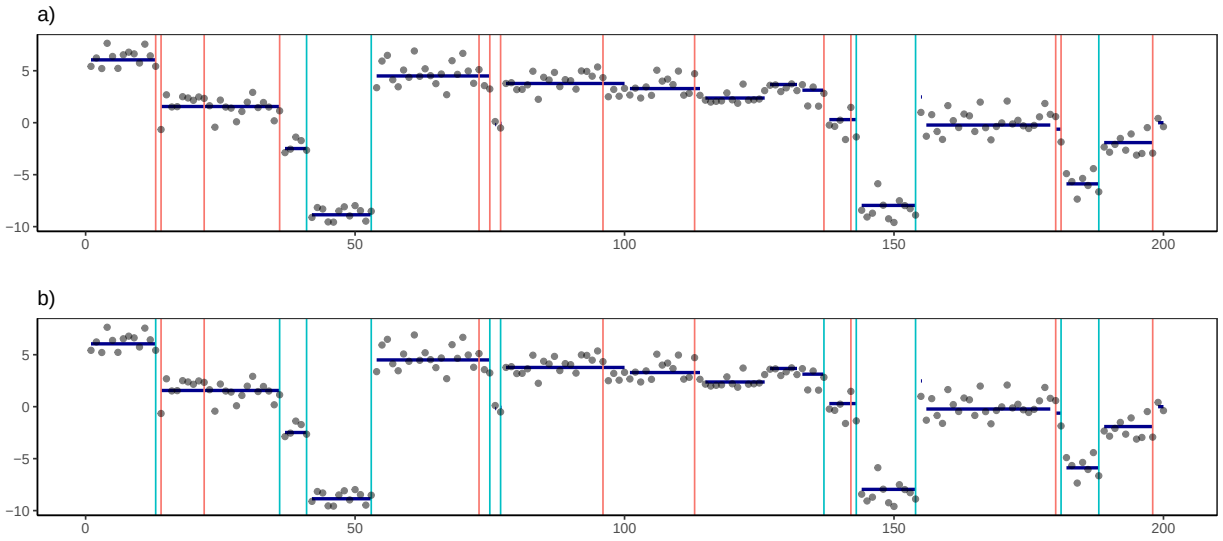


Figure 5.1: The power of a test of (5.9) critically depends on the size of the conditioning set. Observations (displayed in grey) were simulated from (5.1) with $\sigma = 1$ and $\mu_1, \ldots, \mu_T$ displayed in dark blue. Our proposed test of (5.9) was conducted for each of the changepoints estimated via 19-step binary segmentation. Estimated changepoints for which the p-value is less than 0.05 are displayed in blue, and the remaining estimated changepoints are displayed in red. In panel (a), we conducted our proposed test by conditioning on $\mathcal{M}(Y) = \mathcal{M}(y), \mathcal{O}(Y) = \mathcal{O}(y), \Delta(Y) = \Delta(y)$, and $\Pi_\nu^\perp Y = \Pi_\nu^\perp y$ (this is essentially the proposal of Hyun et al. (2018)). In panel (b), we conditioned on the much larger set $\mathcal{M}(Y) = \mathcal{M}(y)$ and $\Pi_\nu^\perp Y = \Pi_\nu^\perp y$.

example, *conditioning on less information allows us to reject the null hypothesis when it is false more often (i.e., we obtain five additional true positives), without inflating the number of false positives.*

With this toy example in mind, we turn to our proposal in the following section. It does not require polyhedral conditioning sets, and thus allows us to condition on much less information than previously possible.

5.3 Two New Tests With Larger Conditioning Sets

In this section, we consider testing a null hypothesis of the form (5.9) using a much larger conditioning set than used by Hyun et al. (2018). Our approach is similar in spirit to the "general recipe" proposed in Section 6 of Liu et al. (2018). We consider two possible forms of the contrast vector ν in Sections 5.3.1 and 5.3.2.

5.3.1 A Test Of No Change In Mean Between Neighboring Changepoints

We first consider testing the null hypothesis (5.9) for ν defined in (5.6). In order to account for the fact that we estimated the changepoints, it is natural to condition on all of the estimated changepoints, $\mathcal{M}(y) = \{\hat{\tau}_1, \ldots, \hat{\tau}_{\hat{K}}\}$. Thus, we define the p-value

$$p \equiv \Pr_{H_0}\left(|\nu^\top Y| \geq |\nu^\top y| \mid \mathcal{M}(Y) = \mathcal{M}(y), \Pi_\nu^\perp Y = \Pi_\nu^\perp y\right). \tag{5.10}$$

As in Hyun et al. (2018), we condition on $\Pi_\nu^\perp Y = \Pi_\nu^\perp y$ for technical reasons; see Appendix C.1. Roughly speaking, (5.10) asks: "Out of all data sets yielding this particular set of changepoints, what is the probability, under the null that there is no changepoint at this location, that the difference in mean between the segments on either side of $\hat{\tau}_j$ is as large as what is observed?"

Our next result reveals that computing (5.10) involves a univariate truncated normal distribution.

Theorem 5.1 *The p-value in* (5.10) *is equal to*

$$p = Pr\left(|\phi| \geq |\nu^\top y| \mid \mathcal{M}(y'(\phi)) = \mathcal{M}(y)\right), \tag{5.11}$$

where $\phi \sim N(0, \|\nu\|^2\sigma^2)$ *and where*

$$y'(\phi) = y - \frac{\nu\nu^\top y}{||\nu||_2^2} + \frac{\nu\phi}{||\nu||_2^2}. \tag{5.12}$$

In light of Theorem 5.1, to evaluate (5.10) we must simply characterize the set

$$\mathcal{S} = \{\phi : \mathcal{M}(y'(\phi)) = \mathcal{M}(y)\}; \tag{5.13}$$

as we will see in Section 5.3.3, this is the set of perturbations of y that result in no change to the estimated changepoints. In Sections 5.4 and 5.5, we do exactly this in the case of binary and ℓ_0 segmentation, respectively. We discuss the fused lasso in Appendix C.4.

5.3.2 A Test Of No Change In Mean Within A Fixed Window Size

We now consider testing the null hypothesis that there is no change in mean in a window $h > 0$ around the jth estimated changepoint,

$$H_0 : \mu_{\hat{\tau}_j - h + 1} = \ldots = \mu_{\hat{\tau}_j} = \ldots = \mu_{\hat{\tau}_j + h}. \tag{5.14}$$

This is a special case of (5.9) for ν defined as

$$\nu_t = \begin{cases} 0 & \text{if } t \leq \hat{\tau}_j - h \text{ or } t > \hat{\tau}_j + h, \\ \frac{1}{h} & \text{if } \hat{\tau}_j - h < t \leq \hat{\tau}_j, \\ -\frac{1}{h} & \text{if } \hat{\tau}_j < t \leq \hat{\tau}_j + h. \end{cases} \tag{5.15}$$

When considering this null hypothesis, it makes sense to condition only on the jth estimated changepoint, leading to a p-value defined as

$$p \equiv \mathrm{Pr}_{H_0}\left(|\nu^\top Y| \geq |\nu^\top y| \mid \hat{\tau}_j \in \mathcal{M}(Y), \Pi_\nu^\perp Y = \Pi_\nu^\perp y\right), \tag{5.16}$$

where once again, we condition on $\Pi_\nu^\perp Y = \Pi_\nu^\perp y$ for technical reasons. Roughly speaking, (5.16) asks: "Out of all data sets yielding a changepoint at $\hat{\tau}_j$, what is the probability, under the null that there is no changepoint at this location, that the difference in mean within a fixed window of $\hat{\tau}_j$ is as large as what is observed?"

The p-values in (5.16) and (5.10) are calculated for slightly different null hypotheses: the null for (5.16) is that there is no changepoint within a distance h of the estimated changepoint, $\hat{\tau}_j$. By contrast, (5.10) tests for no change in mean between the estimated changepoints immediately before and after $\hat{\tau}_j$. Furthermore, (5.16) conditions on less information. We believe that in many applications, the null hypothesis assumed by (5.16) is more natural and informative since it allows a practitioner to specify how accurately they want to detect

changepoint locations, and it avoids rejecting the null due to changes that are arbitrarily far away from $\hat{\tau}_j$. Moreover, the ability to condition on less information intuitively should lead to higher power. If required, the ideas used to calculate (5.16) can be applied to test for the null hypothesis assumed by (5.10), while conditioning on less information. We further investigate these issues in Sections 5.6 and 5.8.1.

Theorem 5.1 can be extended to show that (5.16) is equal to

$$p = \text{Pr}\left(|\phi| \geq |\nu^\top y| \mid \hat{\tau}_j \in \mathcal{M}(y'(\phi))\right), \tag{5.17}$$

where $\phi \sim N(0, \|\nu\|^2\sigma^2)$, and where $y'(\phi)$ was defined in (5.12). Thus, computing the p-value requires characterizing the set

$$\mathcal{S} = \{\phi : \hat{\tau}_j \in \mathcal{M}(y'(\phi))\}; \tag{5.18}$$

this is the set of perturbations of y that result in estimating a changepoint at $\hat{\tau}_j$.

We show in Sections 5.4 and 5.5 that $\mathcal{S}$ can be efficiently characterized for binary and ℓ_0 segmentation. We discuss the fused lasso in Appendix C.4.

5.3.3 Intuition For $y'(\phi)$ And $\mathcal{S}$

To gain intuition for $y'(\phi)$ in (5.12), we consider ν defined in (5.6). We see that

$$y'_t(\phi) \equiv \begin{cases} y_t & \text{if } t \leq \hat{\tau}_{j-1} \text{ or } t > \hat{\tau}_{j+1}, \\ y_t + \frac{\phi - \nu^\top y}{1 + \frac{\hat{\tau}_j - \hat{\tau}_{j-1}}{\hat{\tau}_{j+1} - \hat{\tau}_j}} & \text{if } \hat{\tau}_{j-1} < t \leq \hat{\tau}_j, \\ y_t - \frac{\phi - \nu^\top y}{1 + \frac{\hat{\tau}_{j+1} - \hat{\tau}_j}{\hat{\tau}_j - \hat{\tau}_{j-1}}} & \text{if } \hat{\tau}_j < t \leq \hat{\tau}_{j+1}. \end{cases} \tag{5.19}$$

Thus, $y'_t(\phi)$ is equal to y_t for $t \leq \hat{\tau}_{j-1}$ or $t > \hat{\tau}_{j+1}$, and otherwise equals the observed data perturbed by a function of ϕ around $\hat{\tau}_j$. In other words, we can view $y'(\phi)$ as a perturbation of the observed data y by a quantity proportional to $\phi - \nu^\top y$, within some window of $\hat{\tau}_j$. Furthermore, $\mathcal{S} = \{\phi : \mathcal{M}(y'(\phi)) = \mathcal{M}(y)\}$ is the set of such perturbations that do not affect the set of estimated changepoints.

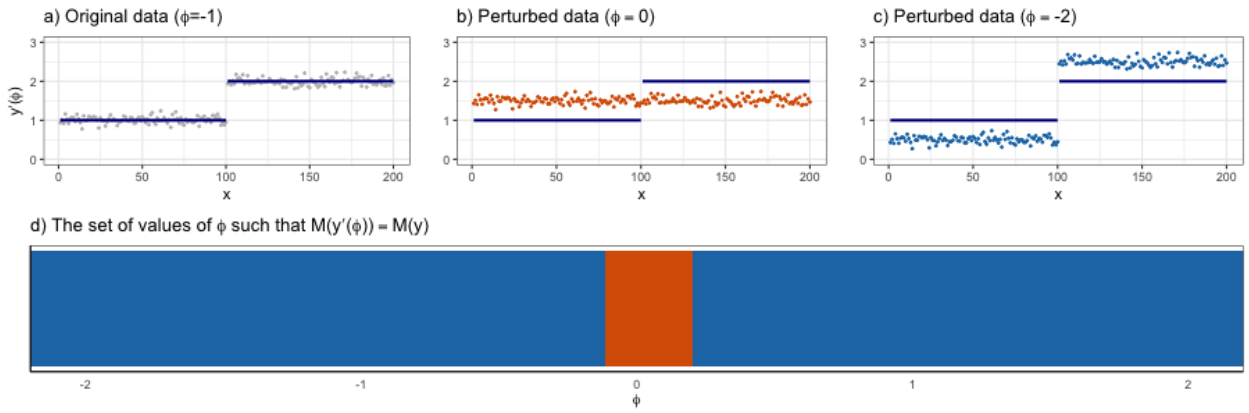


Figure 5.2: a) A simulated dataset with $\phi = \nu^\top y = -1$ is displayed in grey, and the true underlying mean is shown in blue. b) The perturbed dataset $y'(\phi)$ is shown, with $\phi = \nu^\top y = 0$. The perturbed dataset does not have a change in mean at the 100th timepoint, and so 1-step binary segmentation does not detect a changepoint at that position. c) The perturbed dataset $y'(\phi)$ is shown, with $\phi = \nu^\top y = -2$. There is now a very pronounced change in mean at the 100th timepoint, and so 1-step binary segmentation does detect a changepoint at that position. d) Values of ϕ for which $\mathcal{M}(y'(\phi)) = \mathcal{M}(y)$ are shown in blue, and those for which $\mathcal{M}(y'(\phi)) \neq \mathcal{M}(y)$ are shown in red for $\mathcal{M}$ given by 1-step binary segmentation.

Figure 5.2 illustrates the intuition behind $y'(\phi)$ in a simulated example with a change in mean at the 100th position, and where $\phi = \nu^\top y = -1$. In panel a), the observed data are displayed. Here, 1-step binary segmentation estimates $\hat{\tau}_1 = 100$. In panel b), the observed data are perturbed using $\phi = 0$ so that 1-step binary segmentation no longer estimates a changepoint at the 100th position. Conversely, in panel c), the data are perturbed using $\phi = -2$ to exaggerate the change at timepoint 100; 1-step binary segmentation again estimates a changepoint at the 100th position. Hence, for 1-step binary segmentation, -1 and -2 are in $\mathcal{S} = \{\phi : \mathcal{M}(y'(\phi)) = \mathcal{M}(y)\}$, but 0 is not.

In Sections 5.4 and 5.5, and in Appendix C.4, we develop procedures to characterize $\mathcal{S}$ in

the cases of binary segmentation, ℓ_0 segmentation, and the fused lasso, respectively. Here, the procedure from Section 5.4 gives $\mathcal{S} = \{\phi : \mathcal{M}(y'(\phi)) = \mathcal{M}(y)\} = (-\infty, -0.2) \cup (0.2, \infty)$; see panel d) of Figure 5.2.

5.4 Efficient Characterization Of (5.13) *And* (5.18) *For Binary Segmentation*

We now turn our attention to computing the set (5.13) for k-step binary segmentation; (5.18) is detailed in Appendix C.2. Extensions to variants of binary segmentation proposed in Olshen et al. (2004) and Fryzlewicz et al. (2014) are straightforward and, for brevity, are not included.

We begin by paraphrasing Proposition 5.1 of Hyun et al. (2018).

Proposition 5.1 (Proposition 1 of Hyun et al. (2018)) *The set of y for which k-step binary segmentation yields a given set of estimated changepoints, orders, and signs is polyhedral, and takes the form $\{y : \mathbf{\Gamma} y \leq 0\}$ for a $k(2T - k - 3) \times T$ matrix $\mathbf{\Gamma}$, which is a function of the estimated changepoints, orders, and signs.*

We will now make use of this result in a new proposition. Recall from Section 5.2.2 that $\mathcal{M}(y)$, $\mathcal{O}(y)$, and $\Delta(y)$ are defined as the set of estimated changepoints, orders, and signs.

Proposition 5.2 *The set $\{\phi : \mathcal{M}(y'(\phi)) = m, \mathcal{O}(y'(\phi)) = o, \Delta(y'(\phi)) = d\}$ is an interval. Furthermore, the set $\mathcal{S}$ defined in* (5.13) *can be written as the union of such intervals,*

$$\mathcal{S} = \{\phi : \mathcal{M}(y'(\phi)) = \mathcal{M}(y)\} = \bigcup_{i=-N}^{N'} (a_i, a_{i+1}), \tag{5.20}$$

where $N' + N + 1$ is the number of elements in the set

$$\mathcal{I} := \{(o, d) : \exists \alpha \in \mathbb{R} \text{ such that } o = \mathcal{O}(y'(\alpha)), d = \Delta(y'(\alpha)), \mathcal{M}(y) = \mathcal{M}(y'(\alpha))\}. \tag{5.21}$$

That is, $\mathcal{I}$ is the set of possible orders and signs of the changepoints that can be obtained via a perturbation of y that yields changepoints $\mathcal{M}(y)$.

Importantly, $\mathcal{I}$ has far fewer than $2^k k!$ elements, which is the total number of possible orders and signs for the k changepoints. The unconventional indexing in Proposition 5.2 will become apparent. Proposition 5.3 guarantees that Proposition 5.2 is of practical use.

Proposition 5.3 $\bigcup_{i=-N}^{N'}(a_i, a_{i+1})$ *defined in* (5.20) *can be efficiently computed.*

Proposition 5.3 follows from a simple argument that we outline here. We first run k-step binary segmentation on the data y to obtain estimated changepoints $\mathcal{M}(y)$, orders $\mathcal{O}(y)$, and signs $\Delta(y)$. We then apply the first statement in Proposition 5.2 with $m = \mathcal{M}(y)$, $o = \mathcal{O}(y)$, and $d = \Delta(y)$ to identify the interval $[a_0, a_1]$. By construction, $[a_0, a_1] \subset \mathcal{S}$.

Next, for some small positive value of η, we apply the first statement in Proposition 5.2 with $m = \mathcal{M}(y'(a_1 + \eta))$, $o = \mathcal{O}(y'(a_1 + \eta))$, and $d = \Delta(y'(a_1 + \eta))$ to identify the interval $[a_1, a_2]$. (If the left endpoint of this interval does not equal a_1, then we must repeat with a smaller value of η.) We then check whether $\mathcal{M}(y'(a_1 + \eta)) = \mathcal{M}(y)$; if so, then $[a_1, a_2] \subset \mathcal{S}$, and if not, then $[a_1, a_2] \not\subset \mathcal{S}$. Next, we apply the first statement of Proposition 5.2 with $m = \mathcal{M}(y'(a_2 + \eta))$, $o = \mathcal{O}(y'(a_2 + \eta))$, and $d = \Delta(y'(a_2 + \eta))$ to identify the interval $[a_2, a_3]$. We then determine whether $[a_2, a_3] \subset \mathcal{S}$. We continue in this way until we reach an interval containing ∞. We then repeat this process in the other direction, applying the first statement of Proposition 5.2 with $m = \mathcal{M}(y'(a_0 - \eta))$, $o = \mathcal{O}(y'(a_0 - \eta))$, and $d = \Delta(y'(a_0 - \eta))$, and determining whether the resulting interval $[a_{-1}, a_0]$ belongs to $\mathcal{S}$, until eventually we arrive at an interval containing $-\infty$.

Proposition 5.4 shows that this procedure can be stopped early in order to substantially reduce computational costs, and obtain conservative p-values.

Proposition 5.4 *Let* $\tilde{\mathcal{S}}$ *be defined as the set*

$$\tilde{\mathcal{S}} = (-\infty, a_{-r}) \cup \left\{ \bigcup_{i=-r}^{r'} (a_i, a_{i+1}) \right\} \cup (a_{r'+1}, \infty),$$

for some $r < N$ *and* $r' < N'$*, and for* $a_{-r} \leq -|\nu^\top y|$ *and* $a_{r'+1} \geq |\nu^\top y|$*. Then the p-value obtained by conditioning on* $\{\phi \in \tilde{\mathcal{S}}\}$ *is greater than the p-value obtained by conditioning on*

$\{\phi \in \mathcal{S}\}$, *i.e.*,

$$Pr(|\phi| \geq |\nu^\top y| \mid \phi \in \tilde{\mathcal{S}}) \geq Pr(|\phi| \geq |\nu^\top y| \mid \phi \in \mathcal{S}).$$

Appendix C.2 contains proofs of Propositions 5.2 and 5.4. In that section, we also show that Propositions 5.2 and 5.3 can be easily modified to characterize (5.18). Appendix C.4 contains a straightforward modification of this procedure to characterize (5.13) and (5.18) in the case of the fused lasso.

5.5 *Efficient Characterization Of* (5.13) *And* (5.18) *For* ℓ_0 *Segmentation*

In this section, we develop efficient algorithms to analytically characterize (5.13) for the ℓ_0 segmentation problem (5.4) with a fixed value of λ; Appendix C.3 considers $\mathcal{S}$ in the case of (5.18). In particular, we wish to determine all values of ϕ such that $\phi \in \mathcal{S}$, without checking each value of ϕ individually. Roughly speaking, we show that it is possible to write (5.13) in terms of the cost to segment the perturbed data $y'(\phi)$. To compute the necessary cost functions, we derive recursions that look similar to those in Rigaill (2015) and Maidstone et al. (2017b). However these recursions are for functions of two variables, rather than one, which requires fundamentally different techniques to avoid a computational cost that increases exponentially in h.

Let $\hat{K}$ denote the number of estimated changepoints resulting from ℓ_0 segmentation (5.4) on the original data y with fixed tuning parameter value λ, and let $\hat{\tau}_1 < \ldots < \hat{\tau}_{\hat{K}}$ denote the positions of those estimated changepoints; for notational convenience, let $\hat{\tau}_0 \equiv 0$ and $\hat{\tau}_{\hat{K}+1} \equiv T$. Recall the definition of $y'(\phi)$ in (5.12) and the definition of ν in (5.6). For a given value of ϕ, $\mathcal{M}(y'(\phi)) = \mathcal{M}(y)$ if and only if the cost of ℓ_0 segmentation of the data $y'(\phi)$ with the changepoints restricted to occur at $\hat{\tau}_1, \ldots, \hat{\tau}_{\hat{K}}$,

$$C(\phi) = \min_{u_0, u_1, \ldots, u_{\hat{K}}} \left\{ \frac{1}{2} \sum_{k=0}^{\hat{K}} \sum_{t=\hat{\tau}_k+1}^{\hat{\tau}_k} (y'_t(\phi) - u_k)^2 + \lambda \hat{K} \right\}, \tag{5.22}$$

is no greater than the cost of ℓ_0 segmentation of $y'(\phi)$,

$$C'(\phi) = \min_{\substack{0=\tau_0<\tau_1<\cdots<\tau_K<\tau_{K+1}=T,\\ u_0,u_1,\ldots,u_K,K}} \left\{ \frac{1}{2} \sum_{k=0}^{K} \sum_{t=\tau_k+1}^{\tau_{k+1}} (y'_t(\phi) - u_k)^2 + \lambda K \right\}. \tag{5.23}$$

In other words, $\mathcal{S} = \{\phi : C(\phi) \leq C'(\phi)\}$. The following result will prove useful.

Proposition 5.5 *$C(\phi) = C(\phi')$ for all ϕ and ϕ'.*

Proposition 5.5 follows from the fact that from (5.12) and (5.6), $y'(\phi)$ is equal to y_t for $t \leq \hat{\tau}_{j-1}$ or $t > \hat{\tau}_{j+1}$, adds a constant that depends on ϕ to all data points for $\hat{\tau}_{j-1} < t \leq \hat{\tau}_j$, and subtracts a constant from all data points for $\hat{\tau}_j < t \leq \hat{\tau}_{j+1}$. Therefore, by inspection of (5.22), $C(\phi)$ does not depend on the value of ϕ.

Applying Proposition 5.5, we see that $\mathcal{S} = \{\phi : C(\nu^\top y) \leq C'(\phi)\}$. Furthermore, $C(\nu^\top y)$ is easy to calculate, by inspection of (5.22) (recall from (5.12) that $y'(\nu^\top y) = y$). Hence, we simply need an efficient way to calculate $C'(\phi)$, i.e., to perform ℓ_0 segmentation on the perturbed data. In the interest of computational tractability, we need a single procedure that works for all values of ϕ simultaneously, rather than (for instance) having to repeat the procedure for values of ϕ on a fine grid.

We note that $C'(\phi)$ can be decomposed into the cost of segmenting the data $y'(\phi)$ with a changepoint at $\hat{\tau}_j$,

$$C'_{\hat{\tau}_j}(\phi) = \min_u \left\{ \text{Cost}(y'_{1:\hat{\tau}_j}(\phi); u) \right\} + \min_{u'} \left\{ \text{Cost}(y'_{T:(\hat{\tau}_j+1)}(\phi); u') \right\} + \lambda, \tag{5.24}$$

and the cost of segmenting the data $y'(\phi)$ without a changepoint at $\hat{\tau}_j$,

$$C'_{\neg\hat{\tau}_j}(\phi) = \min_u \left\{ \text{Cost}(y'_{1:\hat{\tau}_j}(\phi); u) + \text{Cost}(y'_{T:(\hat{\tau}_j+1)}(\phi); u) \right\}, \tag{5.25}$$

where $\text{Cost}(y'_{1:\hat{\tau}_j}(\phi); u)$ is the cost of segmenting $y'_{1:\hat{\tau}_j}(\phi)$ with $\mu_{\hat{\tau}_j} = u$. Combining (5.24) and (5.25), we have

$$C'(\phi) = \min \left\{ C'_{\hat{\tau}_j}(\phi), C'_{\neg\hat{\tau}_j}(\phi) \right\}. \tag{5.26}$$

Next, we will show that it is possible to analytically calculate $\text{Cost}(y'_{1:\hat{\tau}_j}(\phi); u)$ as a function of the perturbation, ϕ, and the mean at the $\hat{\tau}_j$th timepoint, u. A similar approach can be used to compute $\text{Cost}(y'_{T:(\hat{\tau}_j+1)}(\phi); u)$.

5.5.1 *Analytic Computation Of* $\text{Cost}(y'_{1:\hat{\tau}_j}(\phi); u)$

We first note that $\text{Cost}(y_{1:s}; u)$, the cost of segmenting $y_{1:s}$ with $\mu_s = u$, can be efficiently computed (Rigaill, 2015; Maidstone et al., 2017b). The cost at the first timepoint is simply $\text{Cost}(y_1; u) = \frac{1}{2}(y_1 - u)^2$. For any $s > 1$ and for all u,

$$\text{Cost}(y_{1:s}; u) = \min\Big\{\text{Cost}(y_{1:(s-1)}; u), \min_{u'}\big\{\text{Cost}(y_{1:(s-1)}; u')\big\} + \lambda\Big\} + \frac{1}{2}(y_s - u)^2. \quad (5.27)$$

For each u, this recursion encapsulates two possibilities: (i) there is no changepoint at the $(s-1)$st timepoint, and the optimal cost is equal to the previous cost plus the cost of a new data point, $\text{Cost}(y_{1:(s-1)}; u) + \frac{1}{2}(y_s - u)^2$; (ii) there is a changepoint at the $(s-1)$st timepoint, and the optimal cost is equal to the optimal cost of segmenting up to $s-1$ plus the penalty for adding a changepoint at $s-1$ plus the cost of a new data point, $\min_{u'}\big\{\text{Cost}(y_{1:(s-1)}; u')\big\} + \lambda + \frac{1}{2}(y_s - u)^2$. The resulting cost functions $\text{Cost}(y_1; u), \ldots, \text{Cost}(y_{1:T}; u)$ can be used to determine the exact solution to (5.4). At first blush, the recursion appears to be intractable due to the fact that, naively, $\text{Cost}(y_{1:s}; u)$ needs to updated for each value of $u \in \mathbb{R}$. However, Rigaill (2015) and Maidstone et al. (2017b) show that these updates can be performed by efficiently manipulating piecewise quadratic functions of u, without needing to explicitly consider individual values of u, using a procedure that they call *functional pruning*.

It turns out that many of the computations made in the recursion (5.27) can be reused in the calculation of $\text{Cost}(y'_{1:\hat{\tau}_j}(\phi); u)$. In particular, we note that from (5.12) and (5.6), $y'_s(\phi) = y_s$ for all $s \notin \{\hat{\tau}_{j-1}+1, \ldots, \hat{\tau}_{j+1}\}$, and therefore, $\text{Cost}(y'_{1:\hat{\tau}_{j-1}}(\phi); u) = \text{Cost}(y_{1:\hat{\tau}_{j-1}}; u)$. As a result, we only require a new algorithm to efficiently compute $\text{Cost}(y'_{1:(\hat{\tau}_{j-1}+1)}(\phi); u), \ldots,$ $\text{Cost}(y'_{1:\hat{\tau}_j}(\phi); u)$. However, since these cost functions are piecewise quadratic of two variables, developing functional pruning recursions similar to the one-dimensional recursions of (5.27) is fundamentally more difficult. Nonetheless, in Theorem 5.2 we show that $\text{Cost}(y'_{1:s}(\phi); u)$ for $s = \hat{\tau}_{j-1} + 1, \ldots, \hat{\tau}_j$ is the pointwise minimum over a set $\mathcal{C}_s$ that can be efficiently computed.

Theorem 5.2 *For* $\hat{\tau}_{j-1} < s \leq \hat{\tau}_j$,

$$\text{Cost}(y'_{1:s}(\phi); u) = \min_{f \in \mathcal{C}_s} f(u, \phi),$$

where $\{f(u,\phi)\}_{f\in\mathcal{C}_s}$ *is a collection of* $s-\hat{\tau}_{j-1}+1$ *piecewise quadratic functions of* u *and* ϕ *constructed recursively from* $\hat{\tau}_{j-1}+1$ *to* s*, and where* $\mathcal{C}_{\hat{\tau}_{j-1}}=\{\text{Cost}(y_{1:\hat{\tau}_{j-1}};u)\}$*. Furthermore, the set* $\mathcal{C}_{\hat{\tau}_j}$ *can be computed in* $\mathcal{O}((\hat{\tau}_j-\hat{\tau}_{j-1})^2)$ *operations.*

Appendices C.3 and C.3 contain a proof of Theorem 5.2 and timing results, respectively.

5.5.2 Computing $C'(\phi)$ *Based On* $\text{Cost}(y'_{1:\hat{\tau}_j}(\phi);u)$ *And* $\text{Cost}(y'_{T:(\hat{\tau}_j+1)}(\phi);u)$

Recall from (5.26) that $C'(\phi)$ is the minimum of $C'_{\hat{\tau}_j}(\phi)$ and $C'_{\neg\hat{\tau}_j}(\phi)$, in (5.24) and (5.25), respectively. We now show how to compute $C'_{\hat{\tau}_j}(\phi)$.

We use Theorem 5.2 to build the set $\mathcal{C}_{\hat{\tau}_j}$. Additionally, we define $\tilde{\mathcal{C}}_{\hat{\tau}_{j+1}+1}=\{\text{Cost}(y_{T:(\hat{\tau}_{j+1}+1)};u)\}$, and build $\tilde{\mathcal{C}}_{\hat{\tau}_{j+1}},\ldots,\tilde{\mathcal{C}}_{\hat{\tau}_j+1}$ such that $\text{Cost}(y'_{T:(\hat{\tau}_j+1)}(\phi);u)=\min_{f\in\tilde{\mathcal{C}}_{\hat{\tau}_j+1}}f(u,\phi)$, using a modified version of Theorem 5.2 that accounts for the reversal of the timepoints.

Then, because $\text{Cost}(y'_{1:\hat{\tau}_j}(\phi);u)=\min_{f\in\mathcal{C}_{\hat{\tau}_j}}f(u,\phi)$ and $\text{Cost}(y'_{T:(\hat{\tau}_j+1)}(\phi);u)=\min_{f\in\tilde{\mathcal{C}}_{\hat{\tau}_j+1}}f(u,\phi)$, we have from (5.24) that

$$C'_{\hat{\tau}_j}(\phi)=\min_u\left\{\min_{f\in\mathcal{C}_{\hat{\tau}_j}}\{f(u,\phi)\}\right\}+\min_{u'}\left\{\min_{f\in\tilde{\mathcal{C}}_{\hat{\tau}_j+1}}\{f(u',\phi)\}\right\}+\lambda \tag{5.28}$$

$$=\min_{f\in\mathcal{C}_{\hat{\tau}_j}}\left\{\min_u\{f(u,\phi)\}\right\}+\min_{f\in\tilde{\mathcal{C}}_{\hat{\tau}_j+1}}\left\{\min_{u'}\{f(u',\phi)\}\right\}+\lambda. \tag{5.29}$$

Since $f(u,\phi)$ is piecewise quadratic in u and ϕ (Theorem 5.2), we see that $\min_u\{f(u,\phi)\}$ is piecewise quadratic in ϕ. Therefore, the operation $\min_{f\in\mathcal{C}_{\hat{\tau}_j}}\left\{\min_u\{f(u,\phi)\}\right\}$ can be efficiently performed using ideas from Rigaill (2015) and Maidstone et al. (2017b), and in turn $C'_{\hat{\tau}_j}(\phi)$ can be efficiently computed. Recall from Theorem 5.2 that the set $\mathcal{C}_{\hat{\tau}_j}$ contains $\hat{\tau}_j-\hat{\tau}_{j-1}+1$ functions and can be computed in $\mathcal{O}((\hat{\tau}_j-\hat{\tau}_{j-1})^2)$ operations. Therefore, computing $C'_{\hat{\tau}_j}(\phi)$ requires $\mathcal{O}((\hat{\tau}_j-\hat{\tau}_{j-1})^2)$ operations to compute $\mathcal{C}_{\hat{\tau}_j}$, followed by performing the operation $\min_u\{f(u,\phi)\}$ a total of $\hat{\tau}_j-\hat{\tau}_{j-1}+1$ times. We can similarly obtain the piecewise quadratic function $C'_{\neg\hat{\tau}_j}(\phi)$ of ϕ. Therefore, we can compute $C'(\phi)$.

5.6 Experiments

5.6.1 Simulation Set-up And Methods For Comparison

We simulate $y_1, \ldots, y_{2000}$ according to (5.1) with $\sigma^2 = 1$. The vector $\mu \in \mathbb{R}^{2000}$ has $K = 50$ changepoints, with absolute difference in mean $\delta = |\mu_{\tau_j+1} - \mu_{\tau_j}|$, for $\delta \in \{0.5, 1.0, 1.5, 2.0, 2.5, 3.0, 3.5, 4.0\}$. Panel a) of Figure 5.3 depicts a realization with $\delta = 3$.

We compare four different procedures for testing for a change in mean at an estimated changepoint, $H_0 : \nu^\top \mu = 0$:

Approach 1. Conditioning on the estimated changepoints, order, and signs, $\{\phi : \mathcal{M}(y'(\phi)) = \mathcal{M}(y), \mathcal{O}(y'(\phi)) = \mathcal{O}(y), \Delta(y'(\phi)) = \Delta(y)\}$, for binary segmentation;

Approach 2. Conditioning on all of the estimated changepoints, $\{\phi : \mathcal{M}(y'(\phi)) = \mathcal{M}(y)\}$, for binary segmentation;

Approach 3. Conditioning on the jth estimated changepoint, $\{\phi : \hat{\tau}_j \in \mathcal{M}(y'(\phi))\}$, for binary segmentation;

Approach 4. Conditioning on the jth estimated changepoint, $\{\phi : \hat{\tau}_j \in \mathcal{M}(y'(\phi))\}$, for ℓ_0 segmentation.

As our aim is to compare the power of Approaches 1–4, we assume the true number of changepoints ($K = 50$) is known— so that both binary segmentation and ℓ_0 segmentation estimate the same (or very similar) number of changepoints. On a given data set, there may not exist a value of λ such that ℓ_0 penalization yields precisely 50 estimated changepoints. To see why, consider the noiseless case with two true segments with different means. In this setting, no matter the value of λ, there can be only zero or one estimated changepoints. We also assume that the underlying noise variance ($\sigma^2 = 1$) is known. In what follows, all results reported are averaged over 100 replicate data sets. Unless stated otherwise, we take the window size for testing (5.14) to be $h = 50$. In Approaches 1–3, we approximate the set $\mathcal{S}$ with $\tilde{\mathcal{S}}$ as described in Proposition 5.4; we take $|a_{-r}| = |a_{r'+1}| = \max(10\sigma||\nu||_2, |\nu^\top y|)$.

In practice, model selection techniques can be used to estimate K (Yao, 1988; Lebarbier, 2005; Arlot et al., 2012). Similarly, one can estimate the noise variance σ^2 based on the data y (Birgé and Massart, 2001; Lebarbier, 2005). We note that all of these model selection approaches, except Yao (1988), can be applied in the heteroskedastic setting. Of course, the p-values in (5.11) and (5.17) do not account for these data-driven estimates.

In Appendix C.5, we present timing results for estimating changepoints as well as computing p-values using Approaches 1–4. Surprisingly, Approach 4 is even faster than Approaches 1–3: the former takes only 15 seconds on a series of length $T = 1000$, whereas Approaches 1–3 take longer because calculating $\mathcal{S}$ in the case of binary segmentation requires manipulating a large set of linear equations.

5.6.2 Type I Error Control Under A Global Null

We take $\delta = 0$, so that $\mu_1 = \ldots = \mu_{2000}$, and examine the p-values obtained from each of the four procedures for testing $H_0 : \nu^\top \mu = 0$ in Section 5.6.1. Panel b) of Figure 5.3 displays quantile-quantile plots of the observed p-value quantiles versus theoretical Unif$[0, 1]$ quantiles. The plots indicate Type I error control.

5.6.3 Increases In Power Due To Conditioning On Less Information

Next, we illustrate that power increases as the size of the conditioning event increases, by considering Approaches 1–3 from Section 5.6.1. Each approach uses binary segmentation; the only difference is in the size of the conditioning sets.

On a given dataset, we define the empirical power as the ratio between the number of true changepoints for which the nearest estimated changepoint has a p-value less than α and is within $\pm m$ timepoints, and the number of true changepoints,

$$\text{Power} := \frac{\sum_{i=1}^{K} 1_{\left(|\tau_i - \hat{\tau}_{j(i)}| \leq m \text{ and } p_{j(i)} \leq \alpha\right)}}{K}. \tag{5.30}$$

Here, $j(i) = \text{argmin}_{1 \leq l \leq K} |\tau_i - \hat{\tau}_l|$. Panel c) of Figure 5.3 shows the empirical power for each of the four approaches with $\alpha = 0.05$ and $m = 2$. As the size of the conditioning set increases,

the power increases substantially: the power increases by up to 15% when we condition on $\{\phi : \mathcal{M}(y'(\phi)) = \mathcal{M}(y)\}$ instead of $\{\phi : \mathcal{M}(y'(\phi)) = \mathcal{M}(y), \mathcal{O}(y'(\phi)) = \mathcal{O}(y), \Delta(y'(\phi)) = \Delta(y)\}$, and it increases by another 20% when we condition on $\{\phi : \hat{\tau}_j \in \mathcal{M}(y'(\phi))\}$ instead of $\{\phi : \mathcal{M}(y'(\phi)) = \mathcal{M}(y)\}$.

5.6.4 Power And Detection Probability

We now compare the performances of Approaches 1–4, defined in Section 5.6.1, as well as two additional approaches that are based on *sample splitting* (Cox, 1975):

> *Approach 5.* Apply binary segmentation on the odd timepoints to estimate changepoints. Then apply a standard z-test of $H_0 : \nu^\top \mu = 0$ on the even timepoints;
>
> *Approach 6.* Apply ℓ_0 segmentation to the odd timepoints to estimate changepoints. Then apply a standard z-test of $H_0 : \nu^\top \mu = 0$ on the even timepoints.

Because sample splitting involves estimating the changepoints on half of the data and testing for a change in mean using the other half of the data, the p-value resulting from a standard z-test for a change in mean is valid, but is conditional on the set of timepoints used to estimate the changepoints (Fithian et al., 2014).

In addition to calculating the empirical power (5.30) for each approach, we also consider each approach's ability to detect the true changepoints. This is defined as the fraction of true changepoints for which there is an estimated changepoint within $\pm m$ timepoints,

$$\text{Detection probability} := \frac{\sum_{i=1}^{K} 1_{\left(\min_{1 \le l \le K} |\tau_i - \hat{\tau}_l| \le m\right)}}{K}. \tag{5.31}$$

Figure 5.4 displays the power and detection probability for Approaches 1–6, where $\alpha = 0.05$ and $m = 2$. In panel a), we see that Approach 4 (which estimates changepoints via ℓ_0 segmentation, and then conditions on only the jth estimated changepoint) yields the highest power, especially for larger values of δ. In panel b), we observe that ℓ_0 segmentation vastly outperforms binary segmentation in terms of its ability to detect true changepoints.

Additionally, Figure 5.4 illustrates the benefit of the inferential framework developed in this chapter over naive sample-splitting approaches. Sample splitting is limited in its ability to detect changepoints, since only half of the data is used to estimate changepoints.

5.6.5 Assessment Of Different Window Sizes For Testing (5.14)

The results in Figure 5.4 suggest that conditioning on just $\hat{\tau}_j \in \mathcal{M}(y'(\phi))$ as in (5.17) yields the greatest power to detect a difference in means around $\hat{\tau}_j$. However, this requires pre-specifying the window size in (5.14). We now address this possible weakness. For window sizes $h \in \{1, 30, 50\}$, we assess the performance of Approaches 3 and 4 from Section 5.6.1 in panel c) of Figure 5.4. We observe that, provided h is large enough, the window size has little effect on the power.

5.7 Real Data Example

We now consider guanine-cytosine (G-C) content on a 2Mb window of human chromosome one, binned so that $T = 2000$. Data was originally accessed from the National Center for Biotechnology Information, and is available via the `R` package `changepoint`.

We estimate changepoints using 20-step binary segmentation, and ℓ_0 segmentation using the penalty $\lambda = 2\hat{\sigma}^2 \log T$, which yields 20 estimated changepoints. Figure 5.5 displays the estimated changepoints from these two methods, along with an indication of whether Approaches 1–4 from Section 5.6.1 resulted in a p-value below 0.05. We see that the number of discoveries (estimated changepoints whose p-value is less than 0.05) increases as the size of the conditioning set increases. In Approach 1 we make 11 discoveries, in Approach 2 we make 13, and in Approaches 3 and 4 we make 15 discoveries.

5.8 Discussion

In this chapter, we show that testing for a change in mean around an estimated changepoint simply requires characterizing the set $\mathcal{S}$, defined in either (5.13) or (5.18). We introduce the necessary computational tools to do this for three popular changepoint detection algorithms.

In the case of ℓ_0 segmentation, we develop new functional pruning recursions that avoid an exponential computational cost. Importantly, since our approach does not rely on the polyhedral lemma of Lee et al. (2016), the conditioning sets that we use are much larger than those in earlier work and lead to higher-powered tests. We now discuss a few extensions of our work.

5.8.1 Larger Conditioning Sets For (5.10)

Similarly to Liu et al. (2018), we note that no special properties of the conditioning set were used in the proof of Theorem 5.1. For instance, instead of conditioning on the full set of changepoints as is done in Section 5.3.1, we could have instead conditioned on the jth estimated changepoint and its immediate neighbors. This would yield a p-value of the form $p = \Pr\left(|\phi| \geq |\nu^\top y| \mid \{\hat{\tau}_{j-1}, \hat{\tau}_j, \hat{\tau}_{j+1}\} \subset \mathcal{M}(y'(\phi))\right)$, and requires only a minor modification to the algorithms in Sections 5.4 and 5.5 and in the Appendices.

For some conditioning sets and changepoint detection algorithms, it might be difficult to characterize $\mathcal{S}$. In this case, it is still possible to approximate $\mathcal{S}$ by testing whether or not $\phi \in \mathcal{S}$ for a fine grid of ϕ values; this approach is also suggested by Liu et al. (2018).

5.8.2 Confidence Intervals For The Change In Mean

To construct confidence intervals for the change in mean, we first define $H_0(c) : \nu^T\mu = c$. We note that since $\mathbb{C}(\phi) = \left\{c \mid \Pr_{H_0(c)}\left(|\phi| \geq |\nu^\top y| \mid \phi \in \mathcal{S}\right) \geq \alpha\right\}$ satisfies $\Pr(\nu^\top\mu \in \mathbb{C}(\phi)|\phi \in \mathcal{S}) \geq 1-\alpha$, the set $\mathbb{C}(\phi)$ is a $100(1-\alpha)\%$ confidence interval for $\nu^\top\mu$. Importantly, we can efficiently calculate $\mathbb{C}(\phi)$ since the set $\mathcal{S}$ is unchanged as we vary c; only the mean of the null distribution for $\nu^\top Y$ changes.

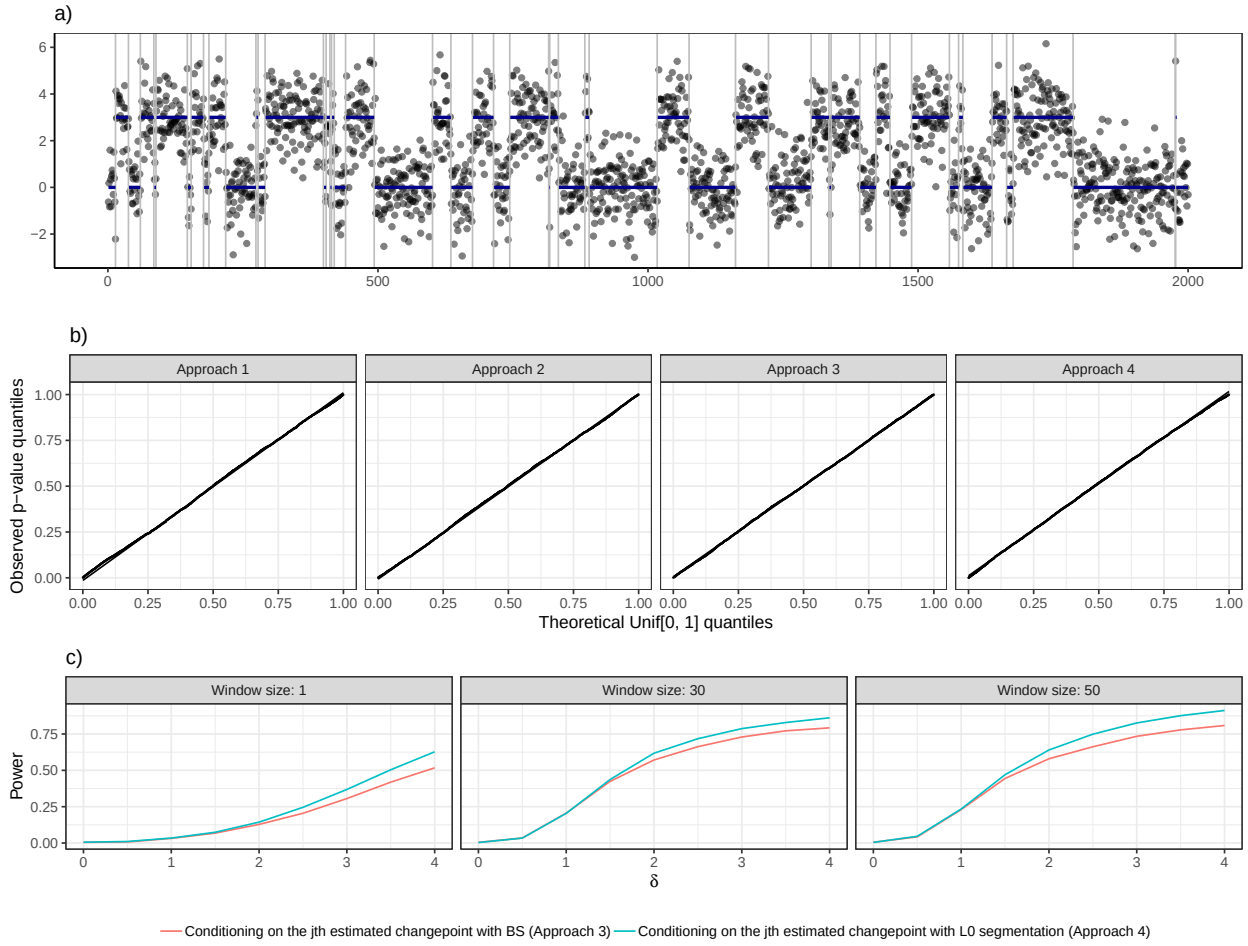


Figure 5.3: a) The grey points represent a realization from the mean model (5.1), with true change in mean due to a changepoint $\delta = 3$. The mean $\mu_1, \ldots, \mu_T$ is shown as a blue line, and the changepoints are shown as grey vertical lines. b) The panels display quantile-quantile plots comparing sample p-value quantiles under (5.1) with $\mu_1 = \ldots = \mu_{2000}$ versus theoretical quantiles of the Unif$(0, 1)$ distribution, for the four approaches listed in Section 5.6.1. c) Empirical power, averaged over 100 replicates, is displayed for Approaches 1–3 defined in Section 5.6.1, each of which results from testing $H_0 : \nu^\top \mu = 0$ for changepoints estimated using binary segmentation with different conditioning sets. Various values of δ, the true change in mean due to a changepoint, are shown on the x-axis. Power increases as the size of the conditioning set increases.

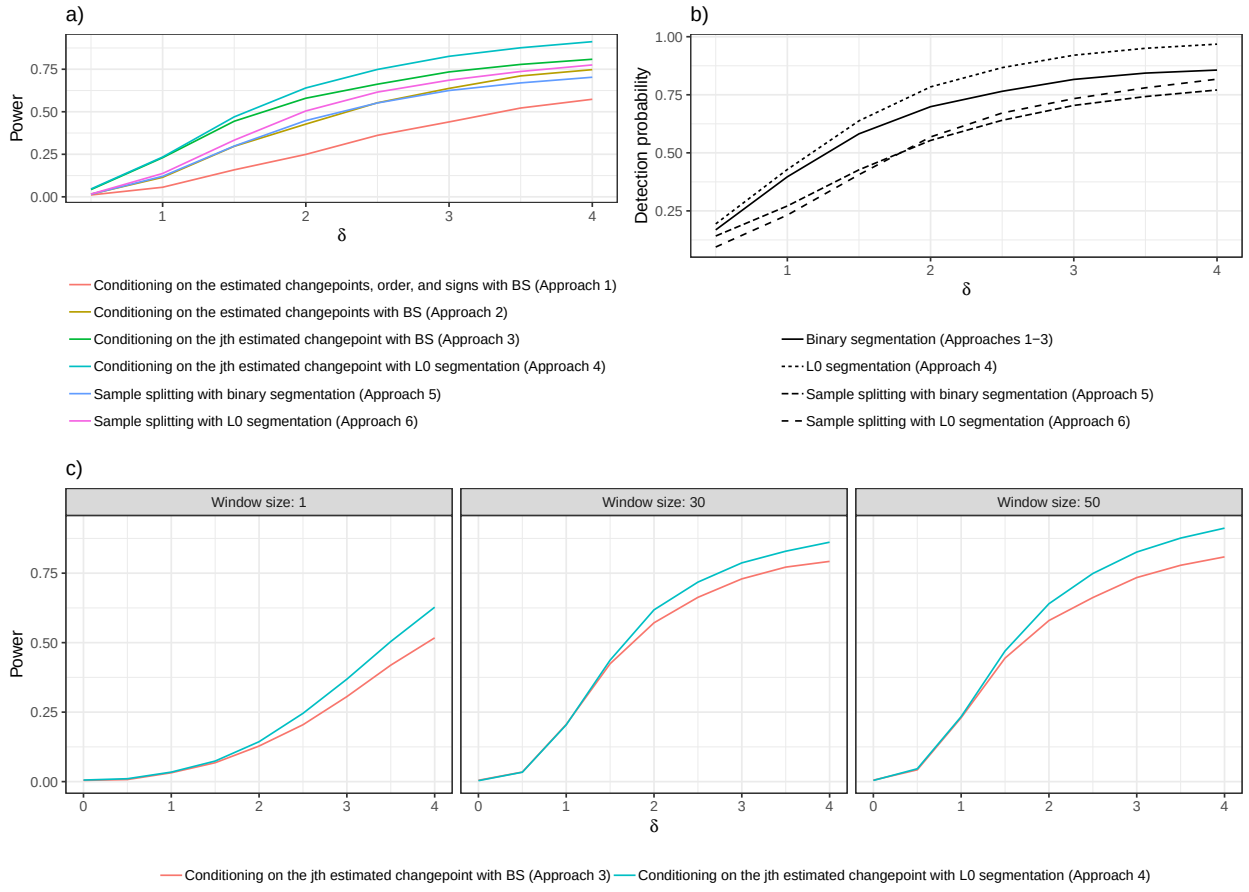


Figure 5.4: Empirical power and detection probability for different changepoint estimation and inference procedures. a) Power for Approaches 1–4, which are described in Section 5.6.1, as well as Approaches 5–6, which are described in Section 5.6.4. b) Detection probability for binary segmentation and ℓ_0 segmentation using all of the data, as well as half of the data. c) Power of Approaches 3 and 4 from Section 5.6.1 for testing the null hypothesis (5.14), for three values of the window size h.

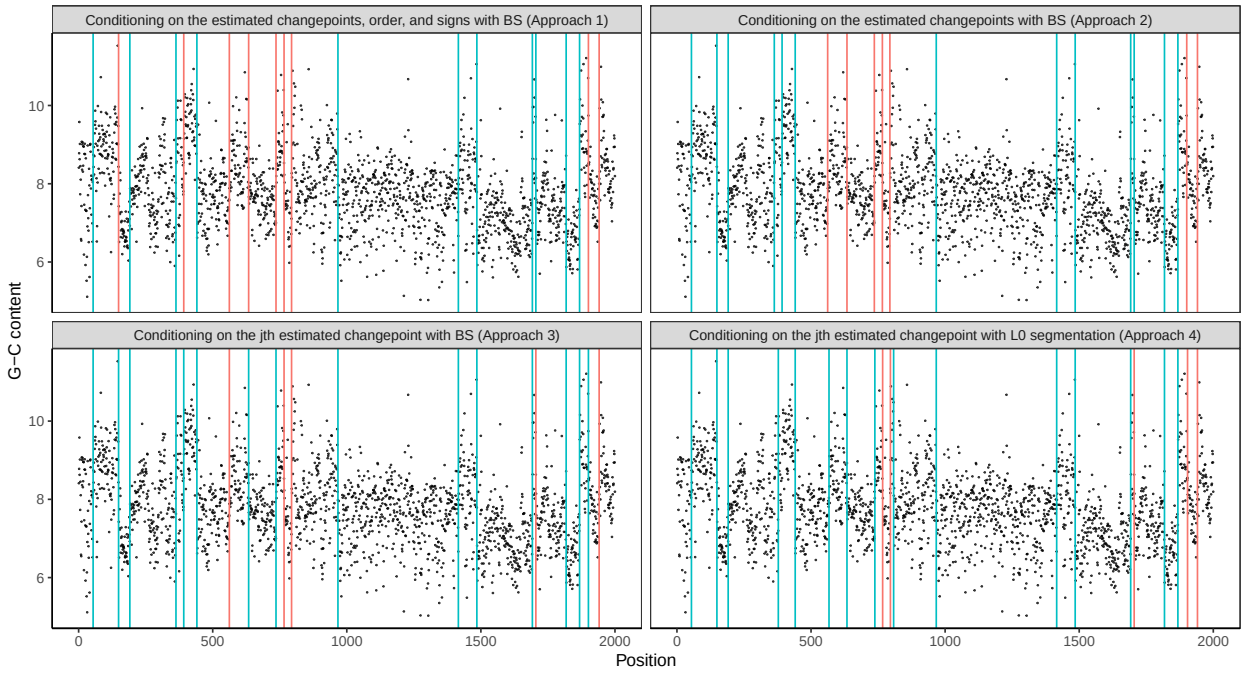


Figure 5.5: The number of discoveries depends on the size of the conditioning set. Scaled G-C content on a 2Mb window of human chromosome one. The G-C content is binned leading to $T = 2000$ (displayed in black). Changepoints are estimated via 20-step binary segmentation, and ℓ_0 segmentation with tuning parameter $\lambda = 2\hat{\sigma}^2 \log(2000) \approx 5.5$. Estimated changepoints from Approaches 1–4 from Section 5.6.1 (organized by panel) for which the p-value is less than 0.05 are displayed in blue; the remaining estimated changepoints are displayed in red.

Chapter 6

DISCUSSION AND FUTURE DIRECTIONS

As described in Chapter 2, neuroscientists are now able to simultaneously record from hundreds to tens of thousands of neurons in behaving animals. This thesis builds upon extensive work to map the fluorescence traces obtained through calcium imaging to the actual times that a neuron is firing (Grewe et al., 2010; Pnevmatikakis et al., 2013; Theis et al., 2016; Deneux et al., 2016; Sasaki et al., 2008; Vogelstein et al., 2009; Yaksi and Friedrich, 2006; Vogelstein et al., 2010; Holekamp et al., 2008; Friedrich and Paninski, 2016; Friedrich et al., 2017). In Chapter 3 we show that the solution to an ℓ_0 optimization problem yields spike estimates that outperform existing approaches. Moreover, we demonstrate that this optimization problem is equivalent to a penalized changepoint detection problem. In Chapter 4, we design a very fast algorithm to solve for the global optimum. This algorithm is used for all of the analyses in the Allen Institute for Brain Science's platform paper (de Vries et al., 2020). Although the spike estimates from Chapter 3 and Chapter 4 exhibit excellent performance, since the estimates arise as the solution to a nonconvex optimization problem, it is challenging to quantify the uncertainty in these point estimates. This motivates Chapter 5, where we introduce a new framework to assess the uncertainty in estimated changepoints post-detection.

This thesis lays the groundwork for several exciting new directions.

6.1 P-values For Spikes Obtained From Calcium Imaging Data

The ideas in Chapter 5 apply beyond the change-in-mean model (5.1). In particular, the ideas in Section 5.3 only require conditioning on the sufficient statistics of $\nu^\top Y$.

For example, we can apply these ideas to analyze data from calcium imaging. Recall that

Vogelstein et al. (2010) and Friedrich et al. (2017) assumed that the observed fluorescence trace for a neuron, y_t, is a noisy version of the underlying calcium concentration, c_t, which decays exponentially with a rate $0 < \gamma < 1$, except when there is an instantaneous increase in the calcium because the neuron has fired, $z_t > 0$:

$$Y_t = c_t + \epsilon_t, \quad \epsilon_t \overset{\text{iid}}{\sim} N(0, \sigma^2), \quad c_t = \gamma c_{t-1} + z_t.$$

In this model, scientific interest lies in determining the times that a neuron fires. In Chapter 3 and Chapter 4, we estimate the spikes by solving

$$\underset{c_1,\ldots,c_T}{\text{minimize}} \left\{ \frac{1}{2} \sum_{t=1}^{T} (y_t - c_t)^2 + \lambda \sum_{t=2}^{T} 1_{(c_t \neq \gamma c_{t-1})} \right\} \text{ subject to } c_t - \gamma c_{t-1} \geq 0,$$

which is closely related to the ℓ_0 segmentation problem (5.4) in Section 5.2.1. We wish to test the null hypothesis that at $\hat{\tau}_j \in \{s : \hat{c}_{s+1} > \gamma \hat{c}_s\}$ there is no increase in the calcium concentration. That is,

$$H_0 : c_{\hat{\tau}_j+1} - \gamma c_{\hat{\tau}_j} = 0, \tag{6.1}$$

versus the one-sided alternative $H_1 : c_{\hat{\tau}_j+1} - \gamma c_{\hat{\tau}_j} > 0$. However, since tests of the form in (6.1) have very little power to detect a change, we propose testing whether there is no increase in the calcium concentration within a window of size h around $\hat{\tau}_j$:

$$H_0 : c^*_{\hat{\tau}_j+1} - \gamma c^*_{\hat{\tau}_j} = 0, \tag{6.2}$$

versus the one-sided alternative $H_1 : c^*_{\hat{\tau}_j+1} - \gamma c^*_{\hat{\tau}_j} > 0$, where

$$c^*_{\hat{\tau}_j} = \underset{c}{\text{argmin}} \left\{ \frac{1}{2} \sum_{t=\hat{\tau}_j-h+1}^{\hat{\tau}_j} \left(c_t - \gamma^{-(\hat{\tau}_j - t)} c\right)^2 \right\}, \text{ and } c^*_{\hat{\tau}_j+1} = \underset{c}{\text{argmin}} \left\{ \frac{1}{2} \sum_{t=\hat{\tau}_j+1}^{\hat{\tau}_j+h} \left(c_t - \gamma^{t-(\hat{\tau}_j+1)} c\right)^2 \right\},$$

The test (6.2) can be written as $H_0 : \nu^\top c = 0$ for a suitable choice of a T-vector contrast ν. The framework from Chapter 5 can be used to test this null hypothesis that there is no increase in the calcium concentration near $\hat{\tau}_j$. Furthermore, the algorithms developed in Chapter 5 can be modified to efficiently characterize the selective distribution.

6.2 Online Estimation And Inference In Changepoint Detection Problems

Another exciting area of future work lies in relaxing an implicit data collection assumption made throughout this thesis. In particular, we assumed that the estimation and inference of changepoints occurs after the whole time series is observed. However, in many settings it is of interest to do estimation and inference of structural changes online; that is, as the data are collected. For example, in neuroscience experiments an investigator may want to choose an action on the basis of a detected neuronal response, or the developer of a wearable device that may want to send a message to the user if a change in behavior is detected.

Indeed, recent work in computational neuroscience has focused on developing online pipelines for calcium imaging data; see Pnevmatikakis (2019) for a review. As one example, Giovannucci et al. (2017) introduce new online methods to identify neurons in calcium imaging videos and subsequently preform near-real time spike estimation on the detected neurons by building on work of Friedrich et al. (2017). It is of interest to explore whether the gains from our techniques developed in Chapters 3, 4, and 5 can be modified for near-real-time estimation and inference.

BIBLIOGRAPHY

Ahrens, M. B., Orger, M. B., Robson, D. N., Li, J. M., and Keller, P. J. (2013). Whole-brain functional imaging at cellular resolution using light-sheet microscopy. *Nature Methods*, 10(5):413–420.

Allen Institute for Brain Science (2016). Stimulus set and response analysis. Technical report, Allen Institute.

Anastasiou, A. and Fryzlewicz, P. (2019). Detecting multiple generalized change-points by isolating single ones. *arXiv preprint arXiv:1901.10852*.

Arlot, S., Celisse, A., and Harchaoui, Z. (2012). A kernel multiple change-point algorithm via model selection. *arXiv preprint arXiv:1202.3878*.

Aue, A. and Horváth, L. (2013). Structural breaks in time series. *Journal of Time Series Analysis*, 34(1):1–16.

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

Badagián, A. L., Kaiser, R., and Peña, D. (2015). Time series segmentation procedures to detect, locate and estimate change-points. In *Empirical Economic and Financial Research*, pages 45–59. Springer.

Bai, J. (1994). Least squares estimation of a shift in linear processes. *Journal of Time Series Analysis*, 15(5):453–472.

Bai, J. and Perron, P. (1998). Estimating and testing linear models with multiple structural changes. *Econometrica*, pages 47–78.

Bai, J. and Perron, P. (2003). Computation and analysis of multiple structural change models. *Journal of Applied Econometrics*, 18(1):1–22.

Barber, R. F., Candès, E. J., et al. (2015). Controlling the false discovery rate via knockoffs. *The Annals of Statistics*, 43(5):2055–2085.

Bauer, P. and Hackl, P. (1980). An extension of the MOSUM technique for quality control. *Technometrics*, 22(1):1–7.

Bellman, R. (1961). On the approximation of curves by line segments using dynamic programming. *Communications of the ACM*, 4(6):284.

Benjamini, Y. and Hochberg, Y. (1995). Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society: Series B (Methodological)*, 57(1):289–300.

Benjamini, Y., Yekutieli, D., et al. (2001). The control of the false discovery rate in multiple testing under dependency. *The Annals of Statistics*, 29(4):1165–1188.

Bien, J. and Witten, D. (2016). *Penalized Estimation in Complex Models, in Handbook of Big Data*, pages 285–303. CRC Press.

Birgé, L. and Massart, P. (2001). A generalized Cp criterion for gaussian model selection.

Boyd, S. and Vandenberghe, L. (2004). *Convex Optimization*. Cambridge University Press.

Boysen, L., Kempe, A., Liebscher, V., Munk, A., and Wittich, O. (2009). Consistencies and rates of convergence of jump-penalized least squares estimators. *The Annals of Statistics*, pages 157–183.

Braun, J. V. and Muller, H.-G. (1998). Statistical methods for dna sequence segmentation. *Statistical Science*, pages 142–162.

Candes, E., Fan, Y., Janson, L., and Lv, J. (2018). Panning for gold:'model-X' knockoffs for high dimensional controlled variable selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 80(3):551–577.

Chen, T.-W., Wardill, T. J., Sun, Y., Pulver, S. R., Renninger, S. L., Baohan, A., Schreiter, E. R., Kerr, R. A., Orger, M. B., Jayaraman, V., et al. (2013). Ultrasensitive fluorescent proteins for imaging neuronal activity. *Nature*, 499(7458):295–300.

Chu, C.-S. J., Hornik, K., and Kaun, C.-M. (1995). MOSUM tests for parameter constancy. *Biometrika*, 82(3):603–617.

Cox, D. R. (1975). A note on data-splitting for the evaluation of significance levels. *Biometrika*, 62(2):441–444.

Dalalyan, A. S., Hebiri, M., Lederer, J., et al. (2017). On the prediction performance of the lasso. *Bernoulli*, 23(1):552–581.

Davies, P. L. and Kovac, A. (2001). Local extremes, runs, strings and multiresolution. *Annals of Statistics*, pages 1–48.

Davis, R. A., Lee, T. C. M., and Rodriguez-Yam, G. A. (2006). Structural break estimation for nonstationary time series models. *Journal of the American Statistical Association*, 101(473):223–239.

de Rooi, J. and Eilers, P. (2011). Deconvolution of pulse trains with the L0 penalty. *Analytica chimica acta*, 705(1):218–226.

de Rooi, J. J., Ruckebusch, C., and Eilers, P. H. (2014). Sparse deconvolution in one and two dimensions: Applications in endocrinology and single-molecule fluorescence imaging. *Analytical chemistry*, 86(13):6291–6298.

de Vries, S. E., Lecoq, J. A., Buice, M. A., Groblewski, P. A., Ocker, G. K., Oliver, M., Feng, D., Cain, N., Ledochowitsch, P., Millman, D., et al. (2020). A large-scale standardized

physiological survey reveals functional organization of the mouse visual cortex. *Nature Neuroscience*, 23(1):138–151.

Deneux, T., Kaszas, A., Szalay, G., Katona, G., Lakner, T., Grinvald, A., Rózsa, B., and Vanzetta, I. (2016). Accurate spike estimation from noisy calcium signals for ultrafast three-dimensional imaging of large neuronal populations in vivo. *Nature Communications*, 7.

Dombeck, D. A., Khabbaz, A. N., Collman, F., Adelman, T. L., and Tank, D. W. (2007). Imaging large-scale neural activity with cellular resolution in awake, mobile mice. *Neuron*, 56(1):43–57.

Duy, V. N. L., Toda, H., Sugiyama, R., and Takeuchi, I. (2020). Computing valid p-value for optimal changepoint by selective inference using dynamic programming. *arXiv preprint arXiv:2002.09132*.

Eichinger, B., Kirch, C., et al. (2018). A MOSUM procedure for the estimation of multiple random change points. *Bernoulli*, 24(1):526–564.

Enikeeva, F. and Harchaoui, Z. (2019). High-dimensional change-point detection under sparse alternatives. *Ann. Statist.*, 47(4):2051–2079.

Fearnhead, P. (2006). Exact and efficient bayesian inference for multiple changepoint problems. *Statistics and computing*, 16(2):203–213.

Fearnhead, P., Maidstone, R., and Letchford, A. (2019). Detecting changes in slope with an L_0 penalty. *Journal of Computational and Graphical Statistics*, 28(2):265–275.

Fisher, W. D. (1958). On grouping for maximum homogeneity. *Journal of the American statistical Association*, 53(284):789–798.

Fithian, W., Sun, D., and Taylor, J. (2014). Optimal inference after model selection. *arXiv preprint arXiv:1410.2597*.

Fithian, W., Taylor, J., Tibshirani, R., and Tibshirani, R. (2015). Selective sequential model selection. *arXiv preprint arXiv:1512.02565*.

Frick, K., Munk, A., and Sieling, H. (2014). Multiscale change point inference. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 76(3):495–580.

Friedrich, J. and Paninski, L. (2016). Fast active set methods for online spike inference from calcium imaging. In *Advances In Neural Information Processing Systems*, pages 1984–1992.

Friedrich, J., Zhou, P., and Paninski, L. (2017). Fast online deconvolution of calcium imaging data. *PLoS Computational Biology*, 13(3):e1005423.

Fryzlewicz, P. et al. (2014). Wild binary segmentation for multiple change-point detection. *The Annals of Statistics*, 42(6):2243–2281.

Futschik, A., Hotz, T., Munk, A., and Sieling, H. (2014). Multiscale DNA partitioning: statistical evidence for segments. *Bioinformatics*, 30(16):2255–2262.

GENIE Project (2015). Simultaneous imaging and loose-seal cell-attached electrical recordings from neurons expressing a variety of genetically encoded calcium indicators. *CRCNS.org*.

Gerstner, W., Kistler, W. M., Naud, R., and Paninski, L. (2014). *Neuronal dynamics: From single neurons to networks and models of cognition*. Cambridge University Press.

Giovannucci, A., Friedrich, J., Kaufman, M., Churchland, A., Chklovskii, D., Paninski, L., and Pnevmatikakis, E. A. (2017). Onacid: Online analysis of calcium imaging data in real time. In *Advances in neural information processing systems*, pages 2381–2391.

Grewe, B. F., Langer, D., Kasper, H., Kampa, B. M., and Helmchen, F. (2010). High-speed in vivo calcium imaging reveals neuronal network activity with near-millisecond precision. *Nature Methods*, 7(5):399–405.

Harchaoui, Z. and Lévy-Leduc, C. (2007). Catching change-points with lasso. In *NIPS*, volume 617, page 624.

Harchaoui, Z. and Lévy-Leduc, C. (2010). Multiple change-point estimation with a total variation penalty. *Journal of the American Statistical Association*, 105(492):1480–1493.

Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The Elements of Statistical Learning; Data Mining, Inference and Prediction*. Springer Verlag, New York.

Hastie, T., Tibshirani, R., and Wainwright, M. (2015). *Statistical Learning with Sparsity*. CRC Press.

Hawrylycz, M., Anastassiou, C., Arkhipov, A., Berg, J., Buice, M., Cain, N., Gouwens, N. W., Gratiy, S., Iyer, R., Lee, J. H., et al. (2016). Inferring cortical function in the mouse visual system through large-scale systems neuroscience. *Proceedings of the National Academy of Sciences*, 113(27):7337–7344.

Haynes, K., Eckley, I. A., and Fearnhead, P. (2017). Computationally efficient changepoint detection for a range of penalties. *Journal of Computational and Graphical Statistics*, 26(1):134–143.

Hinkley, D. V. (1970). Inference about the change-point in a sequence of random variables. *Biometrika*, 57(1):1–17.

Hocking, T. D., Rigaill, G., Fearnhead, P., and Bourque, G. (2017). A log-linear time algorithm for constrained changepoint detection. *arXiv preprint arXiv:1703.03352*.

Hocking, T. D., Rigaill, G., Fearnhead, P., and Bourque, G. (2018). Generalized functional pruning optimal partitioning (gfpop) for constrained changepoint detection in genomic data. *arXiv preprint arXiv:1810.00117*.

Holekamp, T. F., Turaga, D., and Holy, T. E. (2008). Fast three-dimensional fluorescence

imaging of activity in neural populations by objective-coupled planar illumination microscopy. *Neuron*, 57(5):661–672.

Hotz, T., Schütte, O. M., Sieling, H., Polupanow, T., Diederichsen, U., Steinem, C., and Munk, A. (2013). Idealizing ion channel recordings by a jump segmentation multiresolution filter. *IEEE transactions on NanoBioscience*, 12(4):376–386.

Houghton, C. and Kreuz, T. (2012). On the efficient calculation of van rossum distances. *Network: Computation in Neural Systems*, 23(1-2):48–58.

Hugelier, S., de Rooi, J. J., Bernex, R., Duwé, S., Devos, O., Sliwa, M., Dedecker, P., Eilers, P. H., and Ruckebusch, C. (2016). Sparse deconvolution of high-density super-resolution images. *Scientific reports*, 6.

Hušková, M. (1990). Asymptotics for robust MOSUM. *Commentationes Mathematicae Universitatis Carolinae*, 31(2):345–356.

Hyun, S., G'Sell, M., and Tibshirani, R. J. (2016). Exact post-selection inference for changepoint detection and other generalized lasso problems. *arXiv preprint arXiv:1606.03552*.

Hyun, S., Lin, K., G'Sell, M., and Tibshirani, R. J. (2018). Post-selection inference for changepoint detection algorithms with application to copy number variation data. *arXiv preprint arXiv:1812.03644*.

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

James, B., James, K. L., and Siegmund, D. (1987). Tests for a change-point. *Biometrika*, 74(1):71–83.

Jewell, S., Fearnhead, P., and Witten, D. (2019a). Testing for a change in mean after changepoint detection. *arXiv preprint arXiv:1910.04291*.

Jewell, S. and Witten, D. (2018). Exact spike train inference via ℓ_0 optimization. *The Annals of Applied Statistics*, 12(4):2457.

Jewell, S. W., Hocking, T. D., Fearnhead, P., and Witten, D. M. (2019b). Fast nonconvex deconvolution of calcium imaging data. *Biostatistics*. kxy083.

Johnson, N. A. (2013). A dynamic programming algorithm for the fused lasso and ℓ_0-segmentation. *Journal of Computational and Graphical Statistics*, 22(2):246–260.

Killick, R., Fearnhead, P., and Eckley, I. A. (2012). Optimal detection of changepoints with a linear computational cost. *Journal of the American Statistical Association*, 107(500):1590–1598.

Lebarbier, É. (2005). Detecting multiple change-points in the mean of gaussian process by model selection. *Signal Processing*, 85(4):717–736.

Lee, C.-B. (1995). Estimating the number of change points in a sequence of independent normal random variables. *Statistics & Probability Letters*, 25(3):241–248.

Lee, J. D., Sun, D. L., Sun, Y., Taylor, J. E., et al. (2016). Exact post-selection inference, with application to the lasso. *The Annals of Statistics*, 44(3):907–927.

Li, H., Munk, A., Sieling, H., et al. (2016). FDR-control in multiscale change-point segmentation. *Electronic Journal of Statistics*, 10(1):918–959.

Lin, K., Sharpnack, J., Rinaldo, A., and Tibshirani, R. J. (2016). Approximate recovery in changepoint problems, from ℓ_2 estimation error rates. *arXiv preprint arXiv:1606.06746*.

Liu, K., Markovic, J., and Tibshirani, R. (2018). More powerful post-selection inference, with application to the lasso. *arXiv preprint arXiv:1801.09037*.

Ma, T. F. and Yau, C. Y. (2016). A pairwise likelihood-based approach for changepoint detection in multivariate time series models. *Biometrika*, 103(2):409–421.

Maidstone, R., Fearnhead, P., and Letchford, A. (2017a). Detecting changes in slope with an l_0 penalty. *arXiv preprint arXiv:1701.01672*.

Maidstone, R., Hocking, T., Rigaill, G., and Fearnhead, P. (2017b). On optimal multiple changepoint algorithms for large data. *Statistics and Computing*, 27(2):519–533.

Mammen, E., van de Geer, S., et al. (1997). Locally adaptive regression splines. *The Annals of Statistics*, 25(1):387–413.

Muggeo, V. M. and Adelfio, G. (2010). Efficient change point detection for genomic sequences of continuous measurements. *Bioinformatics*, 27(2):161–166.

Nam, C. F., Aston, J. A., and Johansen, A. M. (2012). Quantifying the uncertainty in change points. *Journal of Time Series Analysis*, 33(5):807–823.

Niu, Y. S. and Zhang, H. (2012). The screening and ranking algorithm to detect DNA copy number variations. *The Annals of Applied Statistics*, 6(3):1306.

Olshen, A. B., Venkatraman, E., Lucito, R., and Wigler, M. (2004). Circular binary segmentation for the analysis of array-based dna copy number data. *Biostatistics*, 5(4):557–572.

Pachitariu, M., Stringer, C., and Harris, K. D. (2017). Robustness of spike deconvolution for calcium imaging of neural spiking. *bioRxiv*, page 156786.

Pnevmatikakis, E. A. (2019). Analysis pipelines for calcium imaging data. *Current opinion in neurobiology*, 55:15–21.

Pnevmatikakis, E. A., Merel, J., Pakman, A., and Paninski, L. (2013). Bayesian spike inference from calcium imaging data. In *Signals, Systems and Computers, 2013 Asilomar Conference on*, pages 349–353. IEEE.

Pnevmatikakis, E. A., Soudry, D., Gao, Y., Machado, T. A., Merel, J., Pfau, D., Reardon, T., Mu, Y., Lacefield, C., Yang, W., et al. (2016). Simultaneous denoising, deconvolution, and demixing of calcium imaging data. *Neuron*, 89(2):285–299.

Prevedel, R., Yoon, Y.-G., Hoffmann, M., Pak, N., Wetzstein, G., Kato, S., Schrödel, T., Raskar, R., Zimmer, M., Boyden, E. S., et al. (2014). Simultaneous whole-animal 3D imaging of neuronal activity using light-field microscopy. *Nature Methods*, 11(7):727–730.

Qian, J. and Jia, J. (2012). On pattern recovery of the fused lasso. *arXiv preprint arXiv:1211.5194*.

Quiroga, R. Q. and Panzeri, S. (2009). Extracting information from neuronal populations: information theory and decoding approaches. *Nature Reviews Neuroscience*, 10(3):173.

Reinagel, P. and Reid, R. C. (2000). Temporal coding of visual information in the thalamus. *Journal of Neuroscience*, 20(14):5392–5400.

Rigaill, G. (2015). A pruned dynamic programming algorithm to recover the best segmentations with 1 to k_max change-points. *Journal de la Société Française de Statistique*, 156(4):180–205.

Rojas, C. R. and Wahlberg, B. (2014). On change point detection using the fused lasso method. *arXiv preprint arXiv:1401.5408*.

Ruanaidh, J. J. O. and Fitzgerald, W. J. (2012). *Numerical Bayesian methods applied to signal processing*. Springer Science & Business Media.

Rudin, L. I., Osher, S., and Fatemi, E. (1992). Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena*, 60(1-4):259–268.

Sasaki, T., Takahashi, N., Matsuki, N., and Ikegaya, Y. (2008). Fast and accurate detection of action potentials from somatic calcium fluctuations. *Journal of Neurophysiology*, 100(3):1668–1676.

Schröder, A. L. and Fryzlewicz, P. (2013). Adaptive trend estimation in financial time series via multiscale change-point-induced basis recovery. *Statistics and Its Interface*, 4(6):449–461.

Scott, A. J. and Knott, M. (1974). A cluster analysis method for grouping means in the analysis of variance. *Biometrics*, pages 507–512.

Shen, H. (2016). Brain-data gold mine released: massive survey of mouse visual-cortex activity aims to reveal brain's computational rules. *Nature*, 535(7611):209–211.

Theis, L., Berens, P., Froudarakis, E., Reimer, J., Rosón, M. R., Baden, T., Euler, T., Tolias, A. S., and Bethge, M. (2016). Benchmarking spike rate inference in population calcium imaging. *Neuron*, 90(3):471–482.

Tian, X., Taylor, J., et al. (2018). Selective inference with a randomized response. *The Annals of Statistics*, 46(2):679–710.

Tibshirani, R., Saunders, M., Rosset, S., Zhu, J., and Knight, K. (2005). Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(1):91–108.

Tibshirani, R. J., Taylor, J., et al. (2011). The solution path of the generalized lasso. *The Annals of Statistics*, 39(3):1335–1371.

Tibshirani, R. J., Taylor, J., Lockhart, R., and Tibshirani, R. (2016). Exact post-selection inference for sequential regression procedures. *Journal of the American Statistical Association*, 111(514):600–620.

van Rossum, M. C. (2001). A novel spike distance. *Neural computation*, 13(4):751–763.

Victor, J. D. and Purpura, K. P. (1996). Nature and precision of temporal coding in visual cortex: a metric-space analysis. *Journal of neurophysiology*, 76(2):1310–1326.

Victor, J. D. and Purpura, K. P. (1997). Metric-space analysis of spike trains: theory, algorithms and application. *Network: computation in neural systems*, 8(2):127–164.

Vladimirov, N., Mu, Y., Kawashima, T., Bennett, D. V., Yang, C.-T., Looger, L. L., Keller, P. J., Freeman, J., and Ahrens, M. B. (2014). Light-sheet functional imaging in fictively behaving zebrafish. *Nature Methods*, 11(9):883.

Vogelstein, J. T., Packer, A. M., Machado, T. A., Sippy, T., Babadi, B., Yuste, R., and Paninski, L. (2010). Fast nonnegative deconvolution for spike train inference from population calcium imaging. *Journal of Neurophysiology*, 104(6):3691–3704.

Vogelstein, J. T., Watson, B. O., Packer, A. M., Yuste, R., Jedynak, B., and Paninski, L. (2009). Spike inference from calcium imaging using sequential monte carlo methods. *Biophysical Journal*, 97(2):636–655.

Vostrikova, L. (1981). Detection of the disorder in multidimensional random-processes. *Doklady Akademii Nauk SSSR*, 259(2):270–274.

Xiao, F., Luo, X., Hao, N., Niu, Y. S., Xiao, X., Cai, G., Amos, C. I., and Zhang, H. (2019). An accurate and powerful method for copy number variation detection. *Bioinformatics*.

Yaksi, E. and Friedrich, R. W. (2006). Reconstruction of firing rate changes across neuronal populations by temporally deconvolved ca2+ imaging. *Nature Methods*, 3(5):377–383.

Yao, Y.-C. (1987). Approximating the distribution of the maximum likelihood estimate of the change-point in a sequence of independent random variables. *Ann. Statist.*, 15(3):1321–1328.

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

Yao, Y.-C. and Au, S.-T. (1989). Least-squares estimation of a step function. *Sankhyā: The Indian Journal of Statistics, Series A*, pages 370–381.

Zou, H. (2006). The adaptive lasso and its oracle properties. *Journal of the American Statistical Association*, 101(476):1418–1429.

Appendix A

A.1 Proof Of Proposition 3.1

The first sentence follows by inspection. To establish the second sentence, we observe that the cost

$$\mathcal{D}(y_{a:b}) \equiv \min_{c_a, c_t=\gamma c_{t-1},\, t=a+1,\ldots,b} \left\{ \frac{1}{2} \sum_{t=a}^{b} (y_t - c_t)^2 \right\}$$

can be rewritten by direct substitution of the constraint as

$$\mathcal{D}(y_{a:b}) = \min_{c_a} \left\{ \frac{1}{2} \sum_{t=a}^{b} \left(y_t - \gamma^{t-a} c_a\right)^2 \right\}.$$

This is a least squares problem and is minimized at

$$\hat{c}_a = \frac{\sum_{t=a}^{b} y_t \gamma^{t-a}}{\sum_{t=a}^{b} \gamma^{2(t-a)}},$$

which implies that

$$\mathcal{D}(y_{a:b}) = \frac{1}{2} \sum_{t=a}^{b} \left(y_t - \gamma^{t-a} \hat{c}_a\right)^2,$$

and furthermore that for $a < t \leq b$ the fitted values are $\hat{c}_t = \gamma \hat{c}_{t-1}$. Applying this argument to each segment gives the result stated in Proposition 3.1.

A.2 Proof Of Proposition 3.2

The first equation follows by expanding the square for the final form of $\mathcal{D}(y_{a:b})$ in the proof of Proposition 3.1. Given $\mathcal{D}(y_{a:b})$ we can calculate $\mathcal{D}(y_{a:(b+1)})$ in constant time by storing $\sum_{t=a}^{b} \frac{y_t^2}{2}$ and $\sum_{t=a}^{b} y_t \gamma^{t-a}$, and updating each of these sums for the new data point y_{b+1}; we use a closed form expression to calculate $\sum_{t=a}^{b+1} \gamma^{2(t-a)}$. With each of these quantities stored, $\mathcal{D}(\cdot)$ and $\mathcal{C}(\cdot)$ are updated in constant time.

A.3 Choosing λ And γ

Recall that in (3.4), the parameters λ and γ are unknown. The nonnegative parameter λ controls the trade-off between the number of estimated spike events and the quality of the estimated calcium fit to the observed fluorescence. The parameter $\gamma, 0 < \gamma < 1$, controls the rate of exponential decay of the calcium.

Pnevmatikakis et al. (2013), Friedrich and Paninski (2016), and Friedrich et al. (2017) propose to select the exponential decay parameter γ based on the autocovariance function, and to choose the tuning parameter λ such that $||y - \hat{c}||_2 \leq \sigma\sqrt{T}$, where the standard deviation σ is estimated through the power spectral density of y, and T is the number of timepoints. We refer the reader to Friedrich et al. (2017) and Pnevmatikakis et al. (2016) for additional details.

A.4 A Greedy Approach For Approximating The Solution To A Non-Convex Problem

Friedrich et al. (2017) consider a variant of the optimization problem (3.2),

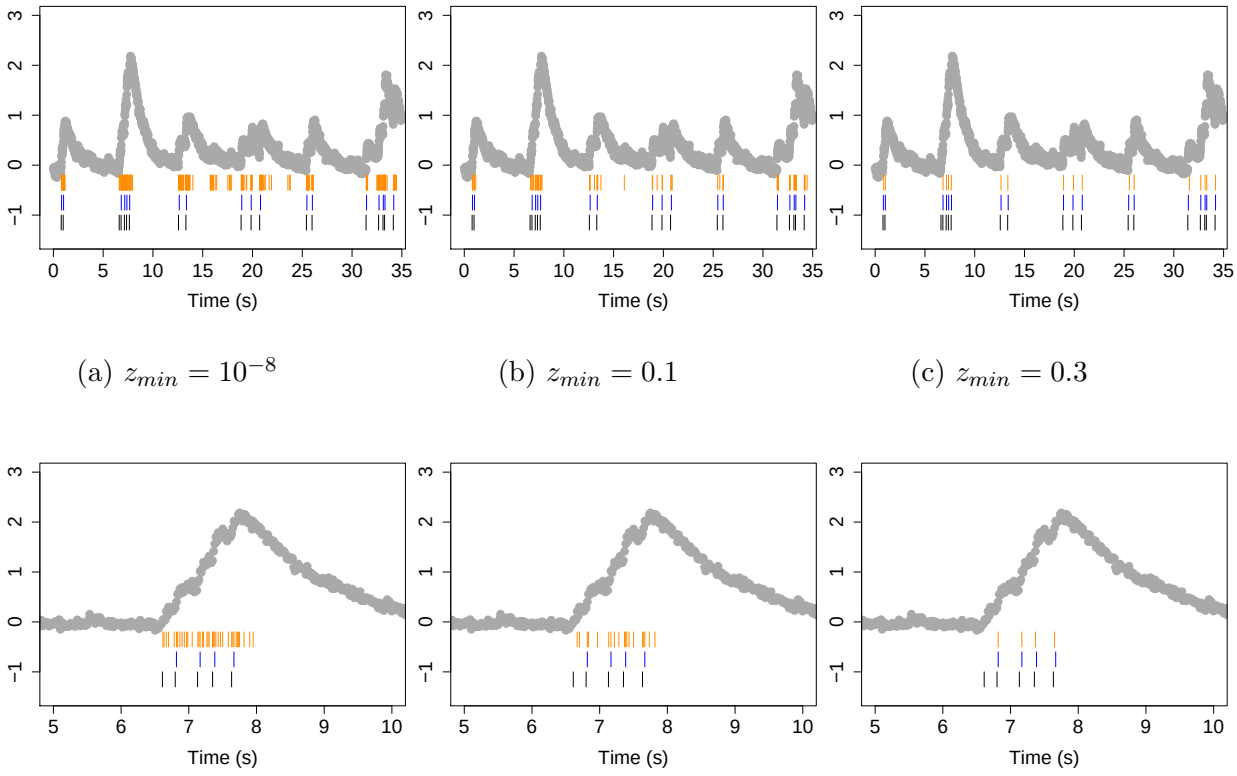
$$\underset{c_1,\ldots,c_T,z_2,\ldots,z_T}{\text{minimize}} \left\{ \frac{1}{2} \sum_{t=1}^{T} (y_t - c_t)^2 \right\} \text{ subject to } z_t = c_t - \gamma c_{t-1} \geq z_{min} \text{ or } z_t = 0, \tag{A.1}$$

obtained from (3.2) by setting $\lambda = 0$, and changing the convex positivity constraint to the non-convex constraint that z_t lies within a non-convex set. Like (3.4), (A.1) is non-convex. Friedrich et al. (2017) do not attempt to solve (A.1) for the global optimum; instead, they provide a heuristic modification to their algorithm for solving (3.4), which is intended to approximate the solution to (A.1).

Figure A.1 illustrates the behavior of this approximate algorithm when applied to the same data as in Figure 3.4. We set $\gamma = 0.9864405$, and considered three values of z_{min}. When $z_{min} = 10^{-8}$ and $z_{min} = 0.1$, in panels (a) to (b), too many spikes are estimated. But when $z_{min} = 0.3$, in panel (c), the solution to (A.1) is very similar to the solution to (3.4) with $\lambda = 0.6$. Both almost perfectly recover the ground truth spikes. Therefore, in this example, the approximate algorithm of Friedrich et al. (2017) for solving (A.1) performs quite well.

However, (A.1) is a non-convex problem, and the approximate algorithm of Friedrich et al. (2017) is not guaranteed to find the global minimum. In fact, we can see that on the data shown in Figure A.1, this approximate algorithm does not find the global optimum. When applied with $z_{min} = 0.3$, the approximate algorithm yields an objective value of 8.57. By contrast, our algorithm for solving (3.4) yields a solution that is feasible for (A.1), and which results in a value of 7.86 for the objective of (A.1). We emphasize that this is quite remarkable: even though the algorithm proposed in Section 3.2 solves (3.4) and not (A.1), *it nonetheless yields a solution that is closer to the global optimum of* (A.1) *than does the approximate algorithm of Friedrich et al. (2017), which is intended to solve* (A.1).

In many cases, the greedy algorithm of Friedrich et al. (2017) for solving (A.1) might yield good results that are near the global optimum of (A.1), and potentially even near the global optimum of (3.4). However, there is no guarantee that this algorithm will yield a "good" local optimum on any given data set. By contrast, in Chapter 3 we proposed an elegant and efficient algorithm for exactly solving the ℓ_0 problem (3.4).



(a) $z_{min} = 10^{-8}$ (b) $z_{min} = 0.1$ (c) $z_{min} = 0.3$

(d) Zoomed $z_{min} = 10^{-8}$ (e) Zoomed $z_{min} = 0.1$ (f) Zoomed $z_{min} = 0.3$

Figure A.1: Spike detection for cell 2002 of the Chen et al. (2013) data. In each panel, the observed fluorescence (•) and true spikes (—) are displayed. Estimated spikes from problem (A.1) are shown in (—), and the estimated spikes from the ℓ_0 problem (3.4) with $\lambda = 0.6$ are shown in (—). Times $0s - 35s$ are shown in the top row; the second row zooms in on times $5s - 10s$ to illustrate behavior around a large increase in calcium concentration. Columns correspond to different values of z_{min}.

Appendix B

B.1 Proof Of Proposition 4.1

The main tool to prove Proposition 4.1 is a simple recursion for the cost function of segmenting the data $y_{1:s}$, given that the most recent changepoint is τ, and that the value of the calcium at the sth timestep is α. For $\tau < s$

$$\begin{aligned}
\text{Cost}_s^\tau(\alpha) &= F(\tau) + \frac{1}{2}\sum_{t=\tau+1}^{s}\left(y_t - \alpha\gamma^{t-s}\right)^2 + \lambda \\
&= \min_{\alpha'} \text{Cost}_\tau^*(\alpha') + \frac{1}{2}\sum_{t=\tau+1}^{s}\left(y_t - \alpha\gamma^{t-s}\right)^2 + \lambda \\
&= \min_{\alpha'} \text{Cost}_\tau^*(\alpha') + \frac{1}{2}\sum_{t=\tau+1}^{s-1}\left(y_t - \alpha\gamma^{t-s}\right)^2 + \lambda + \frac{1}{2}(y_s - \alpha)^2 \\
&= \text{Cost}_{s-1}^\tau(\alpha/\gamma) + \frac{1}{2}(y_s - \alpha)^2. \qquad \text{(B.1)}
\end{aligned}$$

For $\tau = s$, we define

$$\text{Cost}_{s-1}^{s-1}(\alpha/\gamma) \equiv \min_{\alpha'} \text{Cost}_{s-1}^*(\alpha') + \lambda. \qquad \text{(B.2)}$$

Therefore, given the function $\text{Cost}_{s-1}^\tau(\alpha)$, we can form the function $\text{Cost}_s^\tau(\alpha)$.

We now turn our attention to the optimal cost functions $\text{Cost}_s^*(\alpha)$. Consider splitting the minimization over changepoints, $0 \le \tau < s$, in $\text{Cost}_s^*(\alpha)$ as the minimum over changepoints at timesteps $\tau < s-1$ and at $s-1$,

$$\text{Cost}_s^*(\alpha) = \min_{0\le\tau<s} \text{Cost}_s^\tau(\alpha) = \min\left\{\min_{\tau<s-1} \text{Cost}_s^\tau(\alpha), \text{Cost}_s^{s-1}(\alpha)\right\},$$

which combined with the simple recursion in (B.1) gives

$$\begin{aligned}\text{Cost}_s^*(\alpha) &= \min\left\{\min_{\tau<s-1}\text{Cost}_{s-1}^{\tau}(\alpha/\gamma)+\frac{1}{2}(y_s-\alpha)^2, \text{Cost}_{s-1}^{s-1}(\alpha/\gamma)+\frac{1}{2}(y_s-\alpha)^2\right\}\\ &= \min\left\{\text{Cost}_{s-1}^*(\alpha/\gamma), \min_{\alpha'}\text{Cost}_{s-1}^*(\alpha'/\gamma)+\lambda\right\}+\frac{1}{2}(y_s-\alpha)^2\\ &= \min\left\{\text{Cost}_{s-1}^*(\alpha/\gamma), \min_{\alpha'}\text{Cost}_{s-1}^*(\alpha')+\lambda\right\}+\frac{1}{2}(y_s-\alpha)^2.\end{aligned}$$

The second-to-last equality results from the fact that $\text{Cost}_{s-1}^*(\alpha/\gamma) = \min_{\tau<s-1}\text{Cost}_{s-1}^{\tau}(\alpha/\gamma)$ (see (4.5)), and from (B.2). This completes the proof of Proposition 4.1.

B.2 A Fast Functional Pruning Algorithm For Problem (3.3)

Algorithm B.1 efficiently solves (3.3). We notice that Algorithm B.1 is almost identical to Algorithm 4.1. However, there is also a subtle difference as $\text{Cost}_s^{\tau}(\alpha)$ in Algorithm B.1 is defined as in (4.10), whereas in Algorithm 4.1, $\text{Cost}_s^{\tau}(\alpha)$ is defined as in (4.4).

B.3 Implementation Considerations

Slightly Different Optimization Problems That Avoid Numerical Overflow

The recursive updates in (4.6) and (4.9) can give rise to numerical overflow. From (4.6), we see that the recursion connecting $\text{Cost}_s^*(\alpha)$ to $\text{Cost}_{s-1}^*(\alpha/\gamma)$ requires scaling α by a factor of $1/\gamma$. This means that in order to obtain $\text{Cost}_s^*(\alpha)$, we must scale the quadratic coefficients in the piecewise quadratic function $\text{Cost}_{s-1}^*(\alpha)$ by a factor of $1/\gamma^2$, and the linear coefficients by a factor of $1/\gamma$. And for $s<\tau$, in order to obtain $\text{Cost}_s^*(\alpha)$, the quadratic coefficients in $\text{Cost}_{\tau}^*(\alpha)$ must be scaled by a factor of $(1/\gamma^2)^{\tau-s}$. For $\tau-s$ large, this is a large number, leading to numerical overflow.

To avoid numerical overflow, we introduce a new constraint on the minimum value of the calcium, of the form

$$c_t = \max(\gamma c_{t-1}, \epsilon), \tag{B.3}$$

for some small $\epsilon > 0$. In other words, in the absence of a spike, the calcium decreases by a factor of γ in each timestep, until it reaches ϵ, at which point it remains constant until a spike occurs. This constraint suggests two new optimization problems:

$$\underset{c_1,\ldots,c_T}{\text{minimize}} \left\{ \frac{1}{2} \sum_{t=1}^{T} (y_t - c_t)^2 + \lambda \sum_{t=2}^{T} 1_{(c_t \neq \max(\gamma c_{t-1}, \epsilon))} \right\} \tag{B.4}$$

in place of (3.4), and

$$\underset{c_1,\ldots,c_T}{\text{minimize}} \left\{ \frac{1}{2} \sum_{t=1}^{T} (y_t - c_t)^2 + \lambda \sum_{t=2}^{T} 1_{(c_t \neq \max(\gamma c_{t-1}, \epsilon))} \right\} \text{ subject to } c_t \geq \max(\gamma c_{t-1}, \epsilon), \quad t = 2, \ldots, T \tag{B.5}$$

in place of (3.3). For small ϵ, the difference between the old and new formulation is negligible: the objective functions differ only when, under the old formulation, $c_t < \epsilon$, and at such timesteps the difference in objective is bounded by $2y_t\epsilon + \frac{1}{2}\epsilon^2$; in our experiments, we set $\epsilon = 10^{-4}$.

To solve problems (B.4) and (B.5), we redefine $\text{Cost}_s^*(\alpha)$ to account for the minimum calcium concentration constraint (B.3). In particular, we consider two cases: either $c_s = \epsilon$, or $c_s > \epsilon$. The usual updates to $\text{Cost}_s^*(\alpha)$ are valid for $c_s > \epsilon$, whereas a new recursion is needed for case when $c_s = \epsilon$. We describe this recursion next.

First, we consider deriving a recursion to solve (B.4). To calculate the optimal cost of segmenting the data up to time s, conditional on $c_s = \epsilon$, $\text{Cost}_s^*(\epsilon)$, we remark that (B.3) implies two possibilities for the value of c_{s-1}: either $c_{s-1} = \epsilon$ or $\epsilon < c_{s-1} < \epsilon/\gamma$. If $c_{s-1} = \epsilon$, the cost at s is

$$\text{Cost}_s^*(\epsilon) = \text{Cost}_{s-1}^*(\epsilon) + \frac{1}{2}(y_s - \epsilon)^2. \tag{B.6}$$

The second case occurs if $\epsilon < c_{s-1} < \epsilon/\gamma$, as then $c_s = \max(\gamma c_{s-1}, \epsilon) = \epsilon$. In this case, the cost at s is

$$\text{Cost}_s^*(\epsilon) = \min_{\epsilon < \alpha' < \epsilon/\gamma} \text{Cost}_{s-1}^*(\alpha') + \frac{1}{2}(y_s - \epsilon)^2. \tag{B.7}$$

Taken together, (B.6) and (B.7) imply that

$$\text{Cost}_s^*(\epsilon) = \min\left\{\text{Cost}_{s-1}^*(\epsilon), \min_{\epsilon<\alpha'<\epsilon/\gamma} \text{Cost}_{s-1}^*(\alpha')\right\} + \frac{1}{2}(y_s - \epsilon)^2. \tag{B.8}$$

Combining (B.8) with the recursions of (4.6), we arrive at

$$\text{Cost}_s^*(\alpha) = \begin{cases} \min\left\{\text{Cost}_{s-1}^*(\alpha/\gamma), \min_{\alpha'}\text{Cost}_{s-1}^*(\alpha') + \lambda\right\} + \frac{1}{2}(y_s - \alpha)^2 & \alpha > \epsilon, \\ \min\left\{\text{Cost}_{s-1}^*(\epsilon), \min_{\epsilon<\alpha'<\epsilon/\gamma} \text{Cost}_{s-1}^*(\alpha')\right\} + \frac{1}{2}(y_s - \epsilon)^2 & \alpha = \epsilon, \end{cases} \tag{B.9}$$

to solve (B.4). Recursions for (B.5) follow by similar steps.

Recall that the recursions in (4.6) and (4.9) for solving (3.4) and (3.3) result in numerical overflow as a result of the need to scale α by a factor of $1/\gamma$ in $\text{Cost}_{s-1}^*(\alpha/\gamma)$ to obtain $\text{Cost}_s^*(\alpha)$. By contrast, in (B.9), α is no longer scaled for small calcium concentrations. Therefore, (B.9) allows us to solve (B.4) without encountering numerical overflow.

In practice, an algorithm that makes use of (B.9) to solve (B.4) has much lower computational complexity than Algorithm 4.1 and Algorithm B.1, which make use of (4.6) and (4.9) to solve (3.4) and (3.3), respectively. This result is intuitive, as the computational complexity depends on the number of regions needed to represent the cost functions $\text{Cost}_s^*(\alpha)$, and (4.6) and (4.9) result in many regions contained in $[0, \epsilon]$. By contrast, the minimum calcium concentration constraint in (B.4) leads to (B.9), which effectively collapses these regions into a single region. On simulated traces with 100,000 timesteps, the new algorithm is $30 - 80\times$ faster than the old algorithm!

Efficient Implementation Over A Grid Of Tuning Parameter Values

In practice, a practitioner will wish to solve (3.3) and (3.4) over a range of values of the tuning parameter λ, in order to select the optimal value of λ using cross-validation or a related approach. Naively, this requires solving (3.3) and (3.4) separately for each value of λ in a very fine grid. If we can choose a "good" set of λ values—that is, a set of λ values such that each value of λ corresponds to a distinct number of spikes, while also fully covering every possible number of spikes—then we can avoid unnecessary computations.

It turns out that the recent work of Haynes et al. (2017) can be used to efficiently determine how the solution to (3.4) changes as a function of λ. This allows us to solve (3.4) for a prespecified set of well-chosen λ values, instead of needing to conduct a computationally-intensive grid search. We include an implementation of this algorithm in our `FastLZeroSpikeInference` package.

B.4 Enforcing A Minimum Spike Size For The ℓ_0 Problem

As described in Section 4.2.5, the methods used to solve (3.4) can be used to solve the related nonconvex problem (4.13), reproduced here for convenience,

$$\underset{c_1,\ldots,c_T}{\text{minimize}} \left\{ \frac{1}{2}\sum_{t=1}^{T}(y_t - c_t)^2 + \lambda \sum_{t=2}^{T} 1_{\{c_t - \gamma c_{t-1} \neq 0\}} \right\} \text{ subject to } c_t - \gamma c_{t-1} \geq z_{\min} \text{ or } c_t - \gamma c_{t-1} = 0. \tag{B.10}$$

Recall from (4.9) that

$$\text{Cost}_s^*(\alpha) = \min\left\{ \text{Cost}_{s-1}^*(\alpha/\gamma), \min_{\alpha':\alpha \geq \alpha'} \text{Cost}_{s-1}^*(\alpha'/\gamma) + \lambda \right\} + \frac{1}{2}(y_s - \alpha)^2.$$

As stated in Chapter 4, this recursion is *the minimum over two terms, the first of which results from adding an additional point y_s to the current segment,* $\text{Cost}_{s-1}^*(\alpha/\gamma) + \frac{1}{2}(y_s - \alpha)^2$, *and the second of which results from adding a new candidate changepoint at $s-1$ and starting a new segment at timestep s,* $\min_{\alpha':\alpha' \leq \alpha} \text{Cost}_{s-1}^*(\alpha') + \lambda + \frac{1}{2}(y_s - \alpha)^2$.

Our focus is on modifying the second term, $\min_{\alpha':\alpha' \leq \alpha} \text{Cost}_{s-1}^*(\alpha') + \lambda$, since it corresponds to adding a new candidate changepoint. By changing this term to

$$\min_{\alpha':\alpha \geq \alpha' + z_{\min}} \text{Cost}_{s-1}^*(\alpha'/\gamma) + \lambda,$$

we enforce the constraint that a spike can occur at the $(s-1)$th timestep *only* if it results in a calcium increase of at least $z_{\min}$. It is therefore straightforward to solve (4.13) via the recursion

$$\text{Cost}_s^*(\alpha) = \min\left\{ \text{Cost}_{s-1}^*(\alpha/\gamma), \min_{\alpha':\alpha \geq \alpha' + z_{\min}} \text{Cost}_{s-1}^*(\alpha'/\gamma) + \lambda \right\} + \frac{1}{2}(y_s - \alpha)^2. \tag{B.11}$$

In particular, $\min_{\alpha':\alpha\geq\alpha'+z_{\min}} \mathrm{Cost}^*_{s-1}(\alpha'/\gamma)$ can be rewritten, with a simple change of variables, as (4.9)

$$\min_{\alpha':\alpha\geq\alpha'+z_{\min}} \mathrm{Cost}^*_{s-1}(\alpha'/\gamma) = \min_{\alpha'':\alpha\geq\alpha''} \mathrm{Cost}^*_{s-1}((\alpha''-z_{\min})/\gamma),$$

which suggests a straightforward modification of Algorithm B.1.

Recall from (4.7) that the cost functions are piecewise quadratic and can be represented as a list of quadratic, linear, and constant coefficients. The function $\mathrm{Cost}^*_{s-1}((\alpha-z_{\min})/\gamma)$ can be analytically represented through transformations to the coefficients of $\mathrm{Cost}^*_{s-1}(\alpha)$. If $\mathrm{Cost}^*_{s-1}(\alpha) = a\alpha^2+b\alpha+c$, then simple algebra gives $\mathrm{Cost}^*_{s-1}((\alpha-z_{\min})/\gamma) = (a/\gamma^2)\alpha^2 + (b/\gamma - (2az_{\min})/\gamma^2)\alpha + c - (bz_{\min})/\gamma + az^2_{\min}/\gamma^2$. The domain changes from $\alpha\in[l,u]$ to $\alpha\in[\gamma l+z_{\min}, \gamma u+z_{\min}]$.

Algorithm B.2 provides pseudo-code for solving (4.13).

B.5 Example Of Recursion (4.9) *For Solving* (3.3)

Example B.1 *We continue Example 4.1, but this time we use* (4.9) *to solve* (3.3) *rather than using* (4.6) *to solve* (3.4). *Once again, consider the simple dataset* $y=[1.00, 0.98, 0.96, \ldots]$ *with* $\lambda=\frac{1}{2}$ *and* $\gamma=0.98$. *We start with* $\mathrm{Cost}^*_1(\alpha)$, *which is just the quadratic centered around* y_1,

$$\mathrm{Cost}^*_1(\alpha) = \mathrm{Cost}^0_1(\alpha) = \frac{1}{2}(y_1-\alpha)^2 = \frac{1}{2}(1.00-\alpha)^2.$$

To calculate $\mathrm{Cost}^*_2(\alpha)$ *based on* (4.9), *we first calculate*

$$\min_{\alpha'\leq\alpha} \mathrm{Cost}^*_1(\alpha'/\gamma) = \begin{cases} \frac{1}{2}(1-\alpha/\gamma)^2, & 0\leq\alpha<\gamma \\ 0, & \alpha\geq\gamma \end{cases}.$$

We then use (4.9) *to calculate*

$$\begin{aligned}\mathrm{Cost}^*_2(\alpha) &= \min\left\{\mathrm{Cost}^*_1(\alpha/\gamma), \min_{\alpha'\leq\alpha}\mathrm{Cost}^*_1(\alpha'/\gamma)+\lambda\right\} + \frac{1}{2}(y_2-\alpha)^2 \\ &= \begin{cases} \frac{1}{2}(1-\alpha/\gamma)^2 + \frac{1}{2}(0.98-\alpha)^2, & \alpha\in\mathcal{R}^0_2\equiv[0,2\gamma) \\ \frac{1}{2}+\frac{1}{2}(0.98-\alpha)^2, & \alpha\in\mathcal{R}^1_2\equiv[2\gamma,\infty). \end{cases}\end{aligned}$$

Again, to obtain $\text{Cost}_3^*(\alpha)$*, we calculate*

$$\min_{\alpha' \leq \alpha} \text{Cost}_2^*(\alpha'/\gamma) = \begin{cases} \frac{1}{2}(1-\alpha/\gamma^2)^2 + \frac{1}{2}(0.98-\alpha/\gamma)^2, & \alpha \leq \gamma^2 \\ 0, & \alpha \geq \gamma^2 \end{cases}.$$

Then using the recursion (4.9) *we obtain the optimal cost function*

$$\begin{aligned} \text{Cost}_3^*(\alpha) &= \min\left\{\text{Cost}_2^*(\alpha/\gamma), \min_{\alpha' \leq \alpha} \text{Cost}_2^*(\alpha'/\gamma) + \lambda\right\} + \frac{1}{2}(y_3 - \alpha)^2 \\ &= \frac{1}{2}(0.96-\alpha)^2 + \begin{cases} \frac{1}{2}(1-\alpha/\gamma^2)^2 + \frac{1}{2}(0.98-\alpha/\gamma)^2, & \alpha \in \mathcal{R}_3^0 \equiv \left(0, \gamma^2\left(1 + \frac{1}{\sqrt{1+\gamma^2}}\right)\right) \\ \lambda, & \alpha \in \mathcal{R}_3^2 \equiv \left[\gamma^2\left(1 + \frac{1}{\sqrt{1+\gamma^2}}\right), \infty\right) \end{cases}. \end{aligned}$$

Figure B.1 illustrates these updates. This example illustrates that the optimal cost $\text{Cost}_3^*(\alpha)$ *corresponding to* (3.3) *differs from* (3.4)*. In particular, the region corresponding to* $\mathcal{R}_3^0$ *from* (3.3) *is* $\left(0, \gamma^2\left(1 + \frac{1}{\sqrt{1+\gamma^2}}\right)\right)$*, whereas region corresponding to* $\mathcal{R}_3^0$ *from* (3.4) *is* $\gamma^2\left\{\left[0, 1 - \frac{1}{\sqrt{1+\gamma^2}}\right) \cup \left[1 + \frac{1}{\sqrt{1+\gamma^2}}, \infty\right)\right\}$*; compare Figures B.1 and 4.2. However, since the optimal value of* $\text{Cost}_3^*(\alpha)$ *occurs with most recent changepoint zero, the solutions to problems* (3.3) *and* (3.4) *are identical.*

B.6 Computational Complexity Of Solving (3.3) *And* (3.4)

Figure B.2 shows the maximum number of regions, $\max_{s=0,\ldots,T} |\{j : \mathcal{R}_s^j \neq \emptyset, 0 \leq j \leq s-1\}|$ for $\mathcal{R}_s^j$ defined in (4.7), from solving (3.3) and (3.4) with $\lambda = 1$.

B.7 `spikefinder` *Challenge Data*

Table B.1 contains additional information for the `spikefinder` challenge dataset.

B.8 Solving (3.3) *Often Yields Better Estimates Than Solving* (3.4)

Figure B.3 displays the estimated calcium and spike times for the solutions to (3.4) and (3.3) for a short illustrative time window, on just one cell, with tuning parameters set to yield the

true number of spikes on the full 200-second recording. We see that the solution to (3.4) yields a "negative" spike after 170.06 seconds, that is, $\hat{c}_{170.06s} - \gamma\hat{c}_{170.05s} < 0$. By contrast, the solution to (3.3) avoids "negative" spikes.

Algorithm B.1: A functional pruning algorithm for solving (3.3)

Initialize: Compute $\text{Cost}_1^*(\alpha) := \text{Cost}_1^0(\alpha) = \frac{1}{2}(y_1 - \alpha)^2$, and set $\mathcal{R}_1^0 = [0, \infty)$

foreach *timestep* $s = 2, \ldots, T$ **do**

  Calculate and store

  $$\text{Cost}_s^*(\alpha) := \min\{\text{Cost}_{s-1}^*(\alpha/\gamma), \min_{\alpha':\alpha\geq\alpha'} \text{Cost}_{s-1}^*(\alpha'/\gamma) + \lambda\} + \tfrac{1}{2}(y_s - \alpha)^2$$

  Set $\mathcal{R}_s^{s-1} = \{\alpha : \text{Cost}_s^*(\alpha) = \min_{\alpha':\alpha\geq\alpha'} \text{Cost}_{s-1}^*(\alpha'/\gamma) + \lambda + \frac{1}{2}(y_s - \alpha)^2\}$

  foreach $\tau = 0, \ldots, s-1$ **do**

    $\mathcal{R}_s^\tau = (\gamma\mathcal{R}_{s-1}^\tau) \cap (\mathcal{R}_s^{s-1})^c$

  end

end

Perform lines 8-21 of Algorithm 4.1.

Output : Set of changepoints cp, number of changepoints $k := \text{card}(cp)$, and estimated calcium concentrations c.

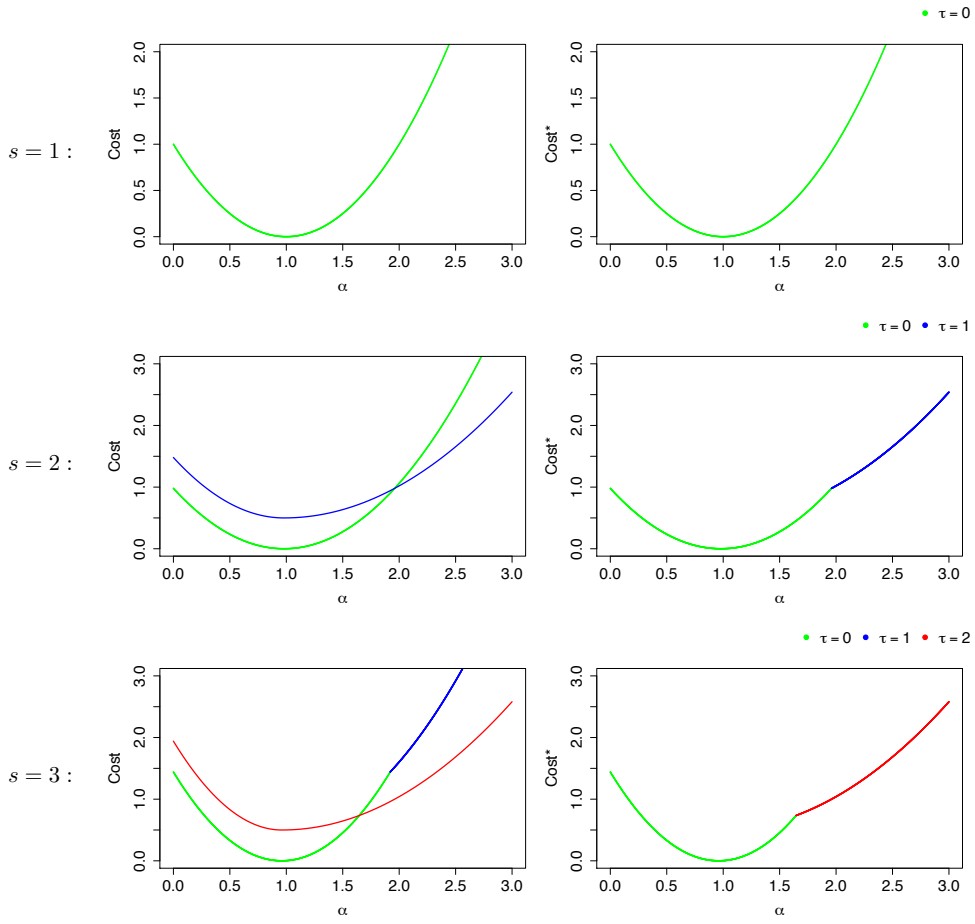


Figure B.1: Evolution of Cost_s^τ and $\text{Cost}_s^*(\alpha)$ for Example B.1. The left-hand panels display the functions $\text{Cost}_{s-1}^*(\alpha/\gamma) + \frac{1}{2}(y_s - \alpha)^2$ and $\min_{\alpha':\alpha' \le \alpha} \text{Cost}_{s-1}^*(\alpha') + \lambda + \frac{1}{2}(y_s - \alpha)^2$, and the right-hand panels show the function $\text{Cost}_s^*(\alpha)$, which is the minimum of those two functions. Rows index the timesteps, $s = 1, 2, 3$. The functions are colored based on the timestep of the most recent changepoint, that is, the value of τ corresponding to $\mathcal{R}_s^\tau$. *Top:* When $s = 1$, $\text{Cost}_1^*(\alpha) = \frac{1}{2}(y_1 - \alpha)^2$; this corresponds to the region $\mathcal{R}_1^0 = [0, \infty)$. *Center:* When $s = 2$, $\text{Cost}_2^*(\alpha)$ is the minimum of two quantities: $\text{Cost}_1^*(\alpha/\gamma) + \frac{1}{2}(y_2 - \alpha)^2$, which corresponds to the most recent changepoint being at timestep zero, and $\min_{\alpha':\alpha' \le \alpha} \text{Cost}_1^*(\alpha') + \lambda + \frac{1}{2}(y_2 - \alpha)^2$, which corresponds to the most recent changepoint being at timestep one. These two functions are shown on the left-hand side, and $\text{Cost}_2^*(\alpha)$ is shown on the right-hand side. *Bottom:* When $s = 3$, $\text{Cost}_3^*(\alpha)$ is calculated similarly; see Example B.1 for additional details.

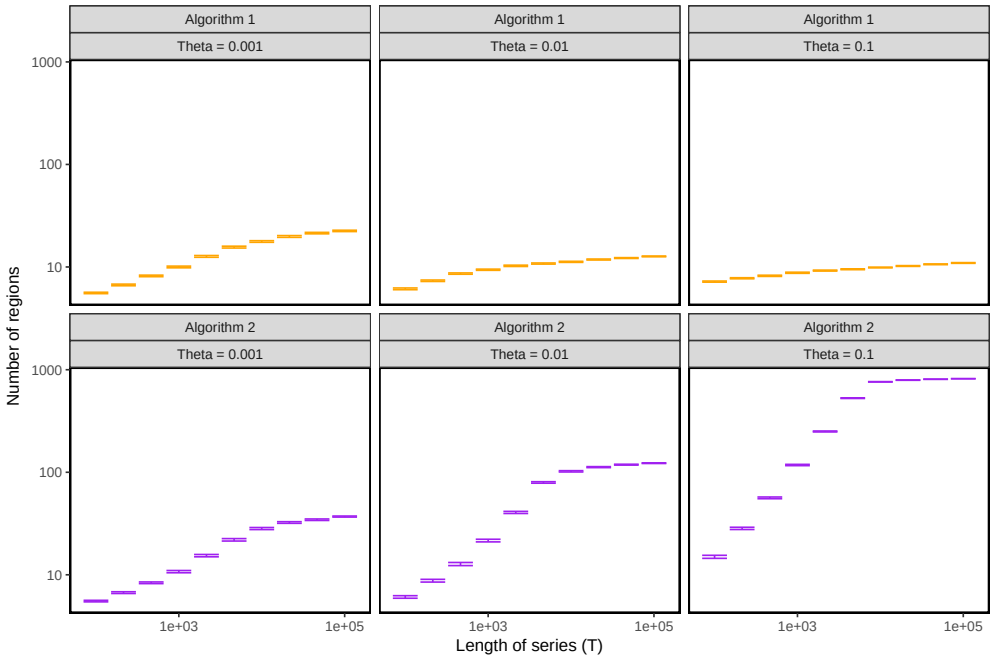


Figure B.2: Maximum number of regions, $\max_{s=0,\ldots,T} |\{j : \mathcal{R}_s^j \neq \emptyset, 0 \leq j \leq s-1\}|$ for $\mathcal{R}_s^j$ defined in (4.7), from solving (3.3) (bottom) and (3.4) (top) with $\lambda = 1$. The panels summarize the results over fifty simulated datasets, each generated according to (2.1) with coefficient $\beta_0 = 0$, $\beta_1 = 1$, decay parameter $\gamma = 0.998$, normal errors $\epsilon_t \overset{\text{ind}}{\sim} \text{N}(0, \sigma = 0.15)$, Poisson distributed spikes $z_t \overset{\text{ind}}{\sim} \text{Pois}(\theta)$ where $\theta \in \{0.1, 0.01, 0.001\}$, and initial calcium value $c_1 \sim \text{Pois}(\theta)$. Panels correspond to different values of θ.

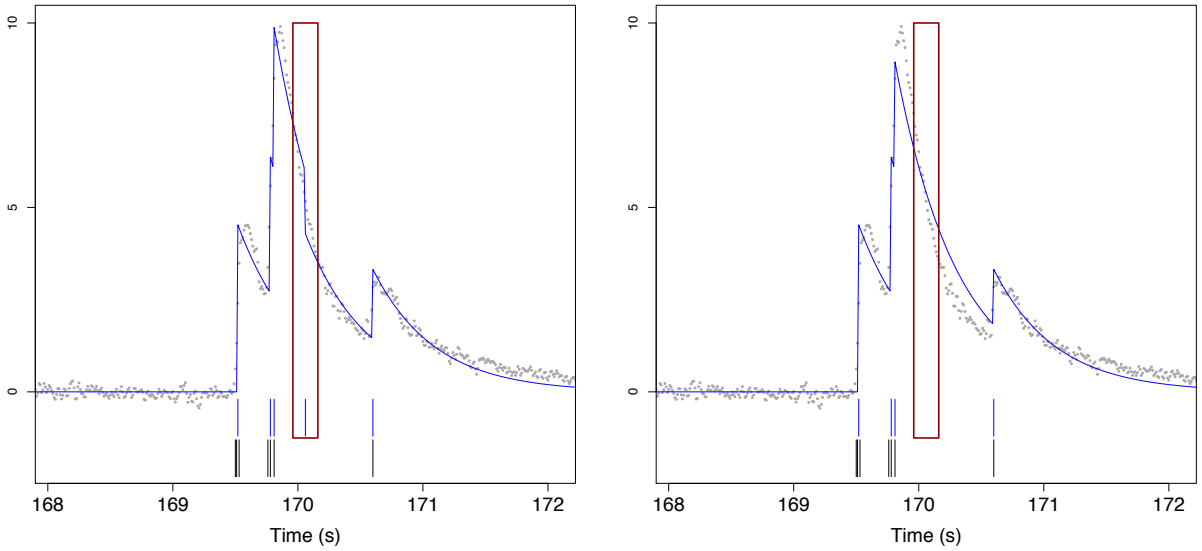


Figure B.3: Illustrative example to show that solving (3.3) often yields better estimates than solving (3.4). Fluorescence and spike train data is from cell 13, GCaMP6f, V1, of Chen et al. (2013) after preprocessing; see Theis et al. (2016). *Left:* Fluorescence trace and true spikes, as well as the calcium and spikes estimated from (3.4). *Right:* Fluorescence trace and true spikes, as well as calcium and spikes estimated from (3.3). In each panel, calcium and estimated spikes are in blue, and true spikes are in black. Tuning parameters set to yield the true number of spike times on the full 200-second recording. The red box highlights the negative spike in the solution to (3.4).

Algorithm B.2: A functional pruning algorithm for solving (4.13)

Initialize: Compute $\text{Cost}_1^*(\alpha) := \text{Cost}_1^0(\alpha) = \frac{1}{2}(y_1 - \alpha)^2$, and set $\mathcal{R}_1^0 = [0, \infty)$

foreach *timestep* $s = 2, \ldots, T$ **do**

  Calculate and store

  $$\text{Cost}_s^*(\alpha) := \min\{\text{Cost}_{s-1}^*(\alpha/\gamma), \min_{\alpha'':\alpha \geq \alpha''} \text{Cost}_{s-1}^*((\alpha'' - z_{\min})/\gamma) + \lambda\} + \tfrac{1}{2}(y_s - \alpha)^2$$

  Set $\mathcal{R}_s^{s-1} = \{\alpha : \text{Cost}_s^*(\alpha) = \min_{\alpha'':\alpha \geq \alpha''} \text{Cost}_{s-1}^*((\alpha'' - z_{\min})/\gamma) + \lambda + \frac{1}{2}(y_s - \alpha)^2\}$

  foreach $\tau = 0, \ldots, s-1$ **do**

    $\mathcal{R}_s^\tau = (\gamma \mathcal{R}_{s-1}^\tau) \cap (\mathcal{R}_s^{s-1})^c$

  end

end

Perform lines 8-21 of Algorithm 4.1.

Output : Set of changepoints cp, number of changepoints $k := \text{card}(cp)$, and estimated calcium concentrations c.

Table B.1: Datasets collected from the `spikefinder` challenge. All datasets were resampled to 100 Hz as part of the challenge.

Indicator	Circuit	Authors	Mean T	Min T	Max T	Num. Cells	Time-scale
OGB-1	V1	Theis et al. 2016	67598	35993	71986	11	Medium
OGB-1	V1	Theis et al. 2016	32285	9682	35508	21	Medium
GCamp6s	V1	Theis et al. 2016	43637	16802	53229	13	Slow
OGB-1	Retina	Theis et al. 2016	27763	27763	27763	6	Medium
GCamp6s	V1	Theis et al. 2016	16919	16919	16919	9	Slow
GCaMP5k	V1	Akerboom et al. 2012	19331	5998	23998	9	Medium
GCaMP6f	V1	Chen et al. 2013	23910	21715	23973	37	Fast
GCaMP6s	V1	Chen et al. 2013	23402	11986	23973	21	Slow
jRCAMP1a	V1	Dana et al. 2016	29460	6486	31959	20	Slow
jRGECO1a	V1	Dana et al. 2016	26086	5507	31994	27	Fast

Appendix C

C.1 Proof Of Theorem 5.1

To characterize (5.10), we note that Y decomposes as

$$Y = (I - \Pi_\nu^\perp)Y + \Pi_\nu^\perp Y, \tag{C.1}$$

where $\Pi_\nu^\perp = I - \frac{\nu\nu^\top}{||\nu||_2^2}$. Then (5.10) becomes

$$p = \mathrm{Pr}_{H_0}\left(|\nu^\top Y| \geq |\nu^\top y| \mid \mathcal{M}(Y) = \mathcal{M}(y), \Pi_\nu^\perp Y = \Pi_\nu^\perp y\right) \tag{C.2}$$

$$= \mathrm{Pr}_{H_0}\left(|\nu^\top Y| \geq |\nu^\top y| \mid \mathcal{M}((I - \Pi_\nu^\perp)Y + \Pi_\nu^\perp y) = \mathcal{M}(y), \Pi_\nu^\perp Y = \Pi_\nu^\perp y\right) \tag{C.3}$$

$$= \mathrm{Pr}_{H_0}\left(|\nu^\top Y| \geq |\nu^\top y| \mid \mathcal{M}((I - \Pi_\nu^\perp)Y + \Pi_\nu^\perp y) = \mathcal{M}(y)\right). \tag{C.4}$$

Here, (C.2) is our definition of a p-value (5.10), and (C.3) follows from (C.1) and the fact that $\Pi_\nu^\perp Y = \Pi_\nu^\perp y$. Finally, (C.4) follows from the fact that Y is Gaussian (see (5.1)) and so $\nu^\top Y$ and $\Pi_\nu^\perp Y$ are independent.

Moreover, we note that (5.1) implies that $\nu^\top Y \sim N(\nu^\top \mu, \|\nu\|^2\sigma^2)$, and that under the null hypothesis (5.9), $\nu^\top Y \sim N(0, \|\nu\|^2\sigma^2)$. We now define $\phi = \nu^\top Y$; thus under the null hypothesis, $\phi \sim N(0, \|\nu\|^2\sigma^2)$. Recall that

$$y'(\phi) = y - \frac{\nu\nu^\top y}{||\nu||_2^2} + \frac{\nu\phi}{||\nu||_2^2}. \tag{C.5}$$

Therefore,

$$p = \mathrm{Pr}\left(|\phi| \geq |\nu^\top y| \mid \mathcal{M}(y'(\phi)) = \mathcal{M}(y)\right). \tag{C.6}$$

C.2 Details Related To Section 5.4

Proof Of Proposition 5.2

To prove the first statement in Proposition 5.2, we note from Proposition 5.1 that the set of data that yields changepoints m, orders o, and signs d is of the form $\{y : \mathbf{\Gamma} y \leq 0\}$.

Therefore, the set of ϕ that yields $\mathcal{M}(y'(\phi)) = m, \mathcal{O}(y'(\phi)) = o$, and $\Delta(y'(\phi)) = d$ is of the form $\{\phi : \mathbf{\Gamma} y'(\phi) \leq 0\}$. Since $\mathbf{\Gamma} y'(\phi) \leq 0$ represents $k(2T - k - 3)$ linear inequalities in ϕ, the set $\{\phi : \mathbf{\Gamma} y'(\phi) \leq 0\}$ is an interval.

The second statement in Proposition 5.2 follows from the fact that

$$\mathcal{S} = \bigcup_{o \in O, d \in D} \{\phi : \mathcal{M}(y'(\phi)) = \mathcal{M}(y), \mathcal{O}(y'(\phi)) = o, \Delta(y'(\phi)) = d\} \tag{C.7}$$

$$= \bigcup_{(o', d') \in \mathcal{I}} \{\phi : \mathcal{M}(y'(\phi)) = \mathcal{M}(y), \mathcal{O}(y'(\phi)) = o', \Delta(y'(\phi)) = d'\} \tag{C.8}$$

$$= \bigcup_{i=-N}^{N'} (a_i, a_{i+1}) \tag{C.9}$$

where O is the set of cardinality $k!$ containing all possible orders of the k changepoints, and $D := \{-1, +1\}^k$ is the set of possible signs. Recall that $N' + N + 1 = |\mathcal{I}|$ for $\mathcal{I}$ defined in (5.21).

A key insight of (C.7)-(C.9) is that (C.7) is the union over $2^k k!$ intervals. By contrast, (C.9) is a union is over $N' + N + 1 = |\mathcal{I}|$ intervals which in practice is much smaller than $2^k k!$.

Proof Of Proposition 5.4

To prove Proposition 5.4, recall that $\mathcal{S} = \bigcup_{i=-N}^{N'} (a_i, a_{i+1})$, as described in Section 5.4, where $a_{-N} = -\infty$ and $a_{N'} = \infty$. Also recall that $\tilde{\mathcal{S}} = (-\infty, a_{-r}) \cup \left\{ \bigcup_{i=-r}^{r'} (a_i, a_{i+1}) \right\} \cup (a_{r'+1}, \infty)$, for some $a_{-r} \leq -|\nu^\top y|$ and $a_{r'+1} \geq |\nu^\top y|$. Since $\{|\phi| \geq |\nu^\top y|\} \cap \{\phi \in \tilde{\mathcal{S}} \setminus \mathcal{S}\} = \{\phi \in \tilde{\mathcal{S}} \setminus \mathcal{S}\}$, we have

$$\begin{aligned}
\Pr(|\phi| \geq |\nu^\top y| \mid \phi \in \tilde{\mathcal{S}}) &= \frac{\Pr(\{|\phi| \geq |\nu^\top y|\} \cap \{\phi \in \tilde{\mathcal{S}}\})}{\Pr(\phi \in \tilde{\mathcal{S}})} \\
&= \frac{\Pr(\{|\phi| \geq |\nu^\top y|\} \cap \{\phi \in \mathcal{S}\}) + \Pr(\{|\phi| \geq |\nu^\top y|\} \cap \{\phi \in \tilde{\mathcal{S}} \setminus \mathcal{S}\})}{\Pr(\phi \in \mathcal{S}) + \Pr(\phi \in \tilde{\mathcal{S}} \setminus \mathcal{S})} \\
&= \frac{\Pr(\{|\phi| \geq |\nu^\top y|\} \cap \{\phi \in \mathcal{S}\}) + \Pr(\phi \in \tilde{\mathcal{S}} \setminus \mathcal{S})}{\Pr(\phi \in \mathcal{S}) + \Pr(\phi \in \tilde{\mathcal{S}} \setminus \mathcal{S})} \\
&\geq \Pr(|\phi| \geq |\nu^\top y| \mid \phi \in \mathcal{S}).
\end{aligned}$$

Characterization Of (5.18)

In this section, we show that we can characterize the set $\mathcal{S} \equiv \{\phi : \hat{\tau}_j \in \mathcal{M}(y'(\phi))\}$ for changepoints estimated via binary segmentation. Our approach is very similar to that of Section 5.4. In the following two propositions, Propositions C.1 and C.2, we modify Propositions 5.2 and 5.3 for the case of $\mathcal{S}$ defined in (5.18).

Proposition C.1 *The set* $\{\phi : \mathcal{M}(y'(\phi)) = m, \mathcal{O}(y'(\phi)) = o, \Delta(y'(\phi)) = d\}$ *is an interval. Furthermore, the set* $\mathcal{S}$ *defined in* (5.18) *can be written as the union of intervals,*

$$\mathcal{S} = \{\phi : \hat{\tau}_j \in \mathcal{M}(y'(\phi))\} = \bigcup_{i=-N}^{N'} (a_i, a_{i+1}), \tag{C.10}$$

where $N' + N + 1$ *is the number of elements in the set*

$$\mathcal{I} := \{(o, d) : \exists \alpha \in \mathbb{R} \text{ such that } o = \mathcal{O}(y'(\alpha)), d = \Delta(y'(\alpha)), \hat{\tau}_j \in \mathcal{M}(y'(\alpha))\}. \tag{C.11}$$

$\mathcal{I}$ *is the set of possible orders and signs of the changepoints that can be obtained via a perturbation of* y *that yields a changepoint at* $\hat{\tau}_j$*.*

Proposition C.2 $\bigcup_{i=-N}^{N'}(a_i, a_{i+1})$ *defined in* (C.10) *can be efficiently computed.*

We outline the proof for Proposition C.2 here. We first run k-step binary segmentation on the data y in order to obtain estimated changepoints $\mathcal{M}(y)$, orders $\mathcal{O}(y)$, and signs $\Delta(y)$. We then apply the first statement in Proposition C.1 to obtain an interval $[a_0, a_1] \subset \mathcal{S}$. Next, for some small positive value of η, we apply the first statement of Proposition C.1 with $m = \mathcal{M}(y'(a_1 + \eta))$, $o = \mathcal{O}(y'(a_1 + \eta))$, and $d = \Delta(y'(a_1 + \eta))$ to identify the interval $[a_1, a_2]$. We then check whether $\hat{\tau}_j \in \mathcal{M}(y'(a_1 + \eta))$; if so, then $[a_1, a_2] \subset \mathcal{S}$, and if not, then $[a_1, a_2] \not\subset \mathcal{S}$. We continue in this vein, much as we did in Section 5.4, to obtain the full set $\mathcal{S}$.

In fact, when characterizing the set $\mathcal{S} = \{\phi : \hat{\tau}_j \in \mathcal{M}(y'(\phi))\}$, this procedure can be sped up. We first define the interval in ϕ such that j-step binary segmentation yields the

estimated changepoints m, orders o, and signs d

$$\{\phi : \mathcal{M}_j(y'(\phi)) = m, \mathcal{O}_j(y'(\phi)) = o, \Delta_j(y'(\phi)) = d\}, \tag{C.12}$$

where the subscripts indicate that we have used j-step binary segmentation as opposed to k-step binary segmentation.

Now, recall that $\hat{\tau}_j$ is the jth estimated changepoint resulting from binary segmentation on the data y. Suppose that $j < k$. We first run j-step binary segmentation on y in order to obtain estimated changepoints $\mathcal{M}_j(y)$, orders $\mathcal{O}_j(y)$, and signs $\Delta_j(y)$. Then we can identify an interval $[a_0, a_1] \subset \mathcal{S}$ by applying (C.12) with $m = \mathcal{M}_j(y)$, $o = \mathcal{O}_j(y)$, and $d = \Delta_j(y)$. This leads to substantial computational speed-ups if $j \ll k$. Next, suppose that $\hat{\tau}_j$ is the lth estimated changepoint resulting from k-step binary segmentation applied to $y'(a_1 + \eta)$, for $l < k$. Once again, we can identify an interval $[a_1, a_2] \subset \mathcal{S}$ by applying (C.12) with $m = \mathcal{M}_l(y'(a_1 + \eta))$, $o = \mathcal{O}_l(y'(a_1 + \eta))$, and $d = \Delta_l(y'(a_1 + \eta))$. By contrast, if $\hat{\tau}_j \notin \mathcal{M}_k(y'(a_1 + \eta))$ or if $\hat{\tau}_j$ is the kth estimated changepoint on the data $y'(a_1 + \eta)$, then we must identify intervals using the first statement of Proposition C.1.

C.3 Details Related To Section 5.5

Proof Of Theorem 5.2

To compute $\text{Cost}(y'_{1:s}(\phi); u)$ for $s \in \{\hat{\tau}_{j-1} + 1, \ldots, \hat{\tau}_{j+1}\}$, we will introduce a set of functions $\mathcal{C}_s$; each function in the set will correspond to a possible configuration for the changepoints preceding the sth timepoint. Then, $\text{Cost}(y'_{1:s}(\phi); u) = \min_{f \in \mathcal{C}_s} f(u, \phi)$. Importantly, we will construct the set $\mathcal{C}_s$ in such a way that its size grows linearly, rather than exponentially, in the size of the set of values that s can take.

To begin, we let $\mathcal{C}_{\hat{\tau}_{j-1}} = \{\text{Cost}(y_{1:\hat{\tau}_{j-1}}; u)\}$ be a set containing a single function, $\text{Cost}(y_{1:\hat{\tau}_{j-1}}; u)$, which can be obtained by applying (5.27) for $s = 1, \ldots, \hat{\tau}_{j-1}$. To obtain the set $\mathcal{C}_{\hat{\tau}_{j-1}+1}$, we must update $\mathcal{C}_{\hat{\tau}_{j-1}}$ to allow for the following two possibilities:

1. *There is no changepoint at the* $(\hat{\tau}_{j-1})$*th timepoint.* In this case, the cost is
$$\text{Cost}(y_{1:\hat{\tau}_{j-1}}; u) + \frac{1}{2}\left(y'_{\hat{\tau}_{j-1}+1}(\phi) - u\right)^2.$$

2. *There is a changepoint at the* $(\hat{\tau}_{j-1})$*th timepoint.* This incurs a penalty of λ, and leads to a cost of
$$\min_{u'}\left\{\text{Cost}(y_{1:\hat{\tau}_{j-1}}; u')\right\} + \frac{1}{2}\left(y'_{\hat{\tau}_{j-1}+1}(\phi) - u\right)^2 + \lambda.$$

Therefore, $\text{Cost}(y'_{1:(\hat{\tau}_{j-1}+1)}(\phi); u) = \min_{f\in\mathcal{C}_{\hat{\tau}_{j-1}+1}} f(u,\phi)$, where

$$\mathcal{C}_{\hat{\tau}_{j-1}+1} = \left\{\text{Cost}(y_{1:\hat{\tau}_{j-1}}; u) + \frac{1}{2}(y'_{\hat{\tau}_{j-1}+1}(\phi) - u)^2,\ \min_{u'}\left\{\text{Cost}(y_{1:\hat{\tau}_{j-1}}; u')\right\} + \frac{1}{2}(y'_{\hat{\tau}_{j-1}+1}(\phi) - u)^2 + \lambda\right\}.$$

Continuing on to the next timepoint, we can see that
$\text{Cost}(y'_{1:(\hat{\tau}_{j-1}+2)}(\phi); u) = \min_{f\in\mathcal{C}_{\hat{\tau}_{j-1}+2}} f(u,\phi)$, where

$$\mathcal{C}_{\hat{\tau}_{j-1}+2} = \Bigg\{\text{Cost}(y_{1:\hat{\tau}_{j-1}}; u) + \frac{1}{2}(y'_{\hat{\tau}_{j-1}+1}(\phi) - u)^2 + \frac{1}{2}(y'_{\hat{\tau}_{j-1}+2}(\phi) - u)^2, \tag{C.13}$$

$$\min_{u'}\left\{\text{Cost}(y_{1:\hat{\tau}_{j-1}}; u')\right\} + \frac{1}{2}(y'_{\hat{\tau}_{j-1}+1}(\phi) - u)^2 + \lambda + \frac{1}{2}(y'_{\hat{\tau}_{j-1}+2}(\phi) - u)^2, \tag{C.14}$$

$$\min_{u''}\left\{\text{Cost}(y_{1:\hat{\tau}_{j-1}}; u'') + \frac{1}{2}(y'_{\hat{\tau}_{j-1}+1}(\phi) - u'')^2\right\} + \frac{1}{2}(y'_{\hat{\tau}_{j-1}+2}(\phi) - u)^2 + \lambda, \tag{C.15}$$

$$\min_{u'}\left\{\text{Cost}(y_{1:\hat{\tau}_{j-1}}; u')\right\} + \min_{u''}\left\{\frac{1}{2}(y'_{\hat{\tau}_{j-1}+1}(\phi) - u'')^2 + \lambda\right\} + \frac{1}{2}(y'_{\hat{\tau}_{j-1}+2}(\phi) - u)^2 + \lambda\Bigg\}. \tag{C.16}$$

Here, (C.13) corresponds to no changepoint at either $\hat{\tau}_{j-1}$ or $\hat{\tau}_{j-1}+1$, (C.14) corresponds to a changepoint at $\hat{\tau}_{j-1}$, (C.15) corresponds to a changepoint at $\hat{\tau}_{j-1}+1$, and (C.16) corresponds to changepoints at $\hat{\tau}_{j-1}$ and $\hat{\tau}_{j-1}+1$. We could continue along this vein to create the sets $\mathcal{C}_{\hat{\tau}_{j-1}+3}, \ldots, \mathcal{C}_{\hat{\tau}_j}$, but the number of functions in the sets would scale exponentially, making computations intractable. Instead, we notice that we really care about the *minimum* of the functions in each set, as a function of u and ϕ; furthermore, since (C.15) and (C.16) are of the form $h(\phi) + \frac{1}{2}(y'_{\hat{\tau}_{j-1}+2}(\phi) - u)^2 + \lambda$, their minimum takes the form

$$\min\left\{\min_{u''}\left\{\text{Cost}(y_{1:\hat{\tau}_{j-1}}; u'') + \frac{1}{2}(y'_{\hat{\tau}_{j-1}+1}(\phi) - u'')^2\right\},\ \min_{u'}\left\{\text{Cost}(y_{1:\hat{\tau}_{j-1}}; u')\right\} + \min_{u''}\left\{\frac{1}{2}(y'_{\hat{\tau}_{j-1}+1}(\phi) - u'')^2 + \lambda\right\}\right\}$$
$$+ \frac{1}{2}(y'_{\hat{\tau}_{j-1}+2}(\phi) - u)^2 + \lambda. \tag{C.17}$$

Thus, it is not necessary for us to keep track of (C.15) and (C.16); we can just keep track of (C.17) instead. Using this insight, as s increases by one, the set $\mathcal{C}_s$ will increase by just one function, rather than increasing exponentially. Importantly, (C.17) is a piecewise quadratic function of ϕ, plus a quadratic function of ϕ and u; therefore, it can be efficiently calculated and stored using ideas from Rigaill (2015) and Maidstone et al. (2017b).

We now summarize the overall procedure. For $s = \hat{\tau}_{j-1} + 1, \ldots, \hat{\tau}_j$, we update the set $\mathcal{C}_s$ as

$$\mathcal{C}_s = \left\{ f(u, \phi) + \frac{1}{2}(y'_s(\phi) - u)^2 : f \in \mathcal{C}_{s-1} \cup \{h_s(\phi)\} \right\}, \tag{C.18}$$

where

$$h_s(\phi) = \min_{f \in \mathcal{C}_{s-1}} \min_{u'} f(u', \phi) + \lambda. \tag{C.19}$$

Furthermore, from (C.18)–(C.19), the size of the set $\mathcal{C}_s$ increases by one as s increases by one. Therefore, computing $\text{Cost}(y'_{1:\hat{\tau}_j}(\phi); u)$ requires $1+2+\ldots+(\hat{\tau}_j - \hat{\tau}_{j-1}) = \mathcal{O}\left((\hat{\tau}_j - \hat{\tau}_{j-1})^2\right)$ operations in the case of (5.13).

Characterization Of (5.18)

In this section, we show that we can characterize the set $\mathcal{S} \equiv \{\phi : \hat{\tau}_j \in \mathcal{M}(y'(\phi))\}$ for changepoints estimated via ℓ_0 segmentation. For $\mathcal{S}$ defined in (5.18), $\phi \in \mathcal{S}$ if and only if the cost of segmenting $y'_{1:T}(\phi)$ with a changepoint at $\hat{\tau}_j$,

$$\tilde{C}(\phi) = \min_u \left\{ \text{Cost}(y'_{1:\hat{\tau}_j}(\phi); u) \right\} + \min_u \left\{ \text{Cost}(y'_{T:(\hat{\tau}_j+1)}(\phi); u) \right\} + \lambda, \tag{C.20}$$

is no greater than the cost of segmenting $y'_{1:T}(\phi)$ with no changepoint at $\hat{\tau}_j$,

$$\tilde{C}'(\phi) = \min_u \left\{ \text{Cost}(y'_{1:\hat{\tau}_j}(\phi); u) + \text{Cost}(y'_{T:(\hat{\tau}_j+1)}(\phi); u) \right\}, \tag{C.21}$$

where $\text{Cost}(y_{1:s}; u)$ is defined in (5.27). Therefore, $\mathcal{S} = \{\phi : \hat{\tau}_j \in \mathcal{M}(y'(\phi))\} = \{\phi : \tilde{C}(\phi) \leq \tilde{C}'(\phi)\}$. We note that (C.20) and (C.21) are identical to (5.24) and (5.25) defined in Section 5.5, except here the contrast ν is defined in (5.15), whereas in Section 5.5 it is defined

in (5.6). Therefore, we can compute $\mathcal{S}$ using a slightly modified version of the procedure of Section 5.5. Section C.3 illustrates the details on a small example.

We also note that computing $\text{Cost}(y'_{1:\hat{\tau}_j}(\phi); u)$ requires $1+2+\ldots+h = \mathcal{O}(h^2)$ operations in the case of (5.18). Timing results are presented in Section C.3.

An Illustration Of The Procedure Of Section C.3

To better grasp the procedure described in Section C.3 to characterize the set $\mathcal{S} = \{\phi : \hat{\tau}_j \in \mathcal{M}(y'(\phi))\}$ in (5.18) for ℓ_0 segmentation, in this section we work through a simple example. Suppose we observe $y = [1, 1, 1, 2, 2, 2]$, and estimate a changepoint at $\hat{\tau} = 3$ by solving (5.4) with $\lambda = \frac{1}{2}$.

In this example, we take $h = 2$, and use the simplified perturbation model

$$y'_t(\phi) = \begin{cases} y_t & t = 1, 6, \\ y_t + \phi & t = 2, 3, \\ y_t - \phi & t = 4, 5. \end{cases} \tag{C.22}$$

We wish to ultimately compute $\mathcal{C}_3$, so we begin with $\mathcal{C}_1 = \{\text{Cost}(y_1; u)\}$,

$$\text{Cost}(y_1; u) = \frac{1}{2}(1-u)^2,$$

and repeatedly use (C.18) and (C.19) to obtain $\mathcal{C}_2$ from $\mathcal{C}_1$ and $\mathcal{C}_3$ from $\mathcal{C}_2$.

$\mathcal{C}_2$ contains two functions: the first function represents the cost of segmenting $[1, 1+\phi]$ with zero changepoints and where the mean $\mu_2 = u$; the second function represents the cost of segmenting $[1, 1+\phi]$ with a changepoint at timepoint 1, and where the mean $\mu_2 = u$. By (C.18), this is simply

$$\mathcal{C}_2 = \left\{ \frac{1}{2}(1-u)^2 + \frac{1}{2}(1+\phi-u)^2, h_2(u, \phi) + \frac{1}{2}(1+\phi-u)^2 \right\},$$

where

$$h_2(u, \phi) = \min_{u'} \text{Cost}(y_1; u') + \lambda = \min_{u'} \frac{1}{2}(1-u')^2 + \frac{1}{2} = \frac{1}{2}.$$

More explicitly,

$$\begin{aligned}\mathcal{C}_2 &= \left\{\frac{1}{2}(1-u)^2 + \frac{1}{2}(y_2'(\phi)-u)^2, \frac{1}{2} + \frac{1}{2}(y_2'(\phi)-u)^2\right\} \\ &= \left\{u^2 - 2u - u\phi + \frac{1}{2}\phi^2 + \phi + 1, \frac{1}{2}u^2 - u - u\phi + \frac{1}{2}\phi^2 + \phi + 1\right\}.\end{aligned}$$

To compute $\mathcal{C}_3$, we first calculate the minimum (corresponding to a changepoint at timepoint 2)

$$h_3(u,\phi) = \min_{f\in\mathcal{C}_2}\min_{u'} f(u',\phi) + \lambda = \begin{cases} 1 & \phi < -\sqrt{2} \\ \frac{1}{4}\phi^2 + \frac{1}{2} & -\sqrt{2} \le \phi \le \sqrt{2} \\ 1 & \phi > \sqrt{2} \end{cases},$$

and add the perturbed data point, $1+\phi$, to obtain $\mathcal{C}_3 = \{q_1(u,\phi), q_2(u,\phi), q_3(u,\phi)\}$, where

$$\begin{aligned} q_1(u,\phi) &= 1.5u^2 - 3u - 2u\phi + \phi^2 + 2\phi + 1.5, \\ q_2(u,\phi) &= u^2 - 2u - 2u\phi + \phi^2 + 2\phi + 1.5, \\ q_3(u,\phi) &= \begin{cases} 0.5u^2 - u - u\phi + 0.5\phi^2 + \phi + 1.5 & \phi < -\sqrt{2} \\ 0.5u^2 - u - u\phi + 0.75\phi^2 + \phi + 1 & -\sqrt{2} \le \phi \le \sqrt{2} \\ 0.5u^2 - u - u\phi + 0.5\phi^2 + \phi + 1.5 & \phi > \sqrt{2} \end{cases} \end{aligned}$$

For any u and ϕ, the optimal cost of segmenting $y'_{1:3}(\phi)$ is given as $\text{Cost}(y'_{1:3}(\phi); u) = \min_{f\in\mathcal{C}_3} f(u,\phi)$.

Applying similar steps in the reverse direction from timepoint 6 to timepoint 4, gives

$$\text{Cost}(y'_{6:4}(\phi); u) = \min\{f_1(u,\phi), f_2(u,\phi), f_3(u,\phi)\},$$

where

$$f_1(u,\phi) = 1.5u^2 - 6u + 2u\phi + \phi^2 - 4\phi + 6,$$
$$f_2(u,\phi) = u^2 - 4u + 2u\phi + \phi^2 - 4\phi + 4.5, \text{ and}$$
$$f_3(u,\phi) = \begin{cases} 0.5u^2 - 2u + u\phi + 0.5\phi^2 - 2\phi + 3 & \phi < -\sqrt{2} \\ 0.5u^2 - 2u + u\phi + 0.75\phi^2 - 2\phi + 2.5 & -\sqrt{2} \le \phi \le \sqrt{2} \\ 0.5u^2 - 2u + u\phi + 0.5\phi^2 - 2\phi + 3 & \phi > \sqrt{2} \end{cases}.$$

$\tilde{C}(\phi)$ and $\tilde{C}'(\phi)$, defined in (C.20) and (C.21), are calculated as

$$\tilde{C}(\phi) = \min_u \text{Cost}(y'_{1:3}(\phi); u) + \min_u \text{Cost}(y'_{6:4}(\phi); u) + \lambda = \begin{cases} \frac{3}{2} & \phi < -\sqrt{\frac{3}{2}} \\ \frac{2}{3}\phi^2 + \frac{1}{2} & -\sqrt{\frac{3}{2}} \le \phi \le \sqrt{\frac{3}{2}} \\ \frac{3}{2} & \phi > \sqrt{\frac{3}{2}} \end{cases},$$

and

$$\tilde{C}'(\phi) = \min_u \{\text{Cost}(y'_{1:3}(\phi); u) + \text{Cost}(y'_{6:4}(\phi)s; u)\}$$
$$= \begin{cases} \phi^2 - \phi + 2.25 & \phi < -1.41421 \\ 1.5\phi^2 - \phi + 1.25 & -1.41421 \le \phi \le -1 \\ 1.625\phi^2 - 1.25\phi + 0.875 & -1 \le \phi \le -0.1547 \\ 2\phi^2 - 2\phi + 0.75 & -0.1547 \le \phi \le 1.76619 \\ 1.375\phi^2 + 1.375\phi + 2.25 & 1.76619 \le \phi \le 1.89681 \\ \phi^2 - \phi + 2.25 & \phi > 1.89681 \end{cases}.$$

To determine $\mathcal{S}$, we recall from Section C.3 that $\mathcal{S} = \{\phi : \tilde{C}(\phi) \le \tilde{C}'(\phi)\}$. Therefore, we

take the minimum

$$\min\left\{\tilde{C}(\phi), \tilde{C}'(\phi)\right\} = \begin{cases} 1.5 & \phi < -1.22474 \quad \text{Minimizer: } \tilde{C}(\phi) \\ \frac{2}{3}\phi + \frac{1}{2} & -1.22474 \leq \phi \leq 0.13763 \quad \text{Minimizer: } \tilde{C}(\phi) \\ 2\phi^2 - 2\phi + 0.75 & 0.13763 \leq \phi \leq 1.29057 \quad \text{Minimizer: } \tilde{C}'(\phi) \\ 1.5 & \phi > 1.29057 \quad \text{Minimizer: } \tilde{C}(\phi) \end{cases}$$

and for each point ϕ track whether $\tilde{C}(\phi)$ or $\tilde{C}'(\phi)$ minimized the objective. Therefore, $\mathcal{S} = (-\infty, 0.13763] \cup [1.29057, \infty)$. Figure C.1 shows $\tilde{C}(\phi)$ and $\tilde{C}'(\phi)$.

Timing Results For Computing The Set $\mathcal{S}$ Defined In (5.18)

In this section, we investigate the claim of Section C.3, that computing the set $\mathcal{S}$ defined in (5.18) in the case of ℓ_0 segmentation requires $\mathcal{O}(h^2)$ computations, where h is the window size that appears in (5.14).

Figure C.2 displays the average running time over 50 replicate datasets as a function of the window size, h, on a simulated dataset of 2000 timepoints, which contains a single changepoint at the 1000th timepoint. We see that the running time is, in fact, approximately quadratic in the window size.

C.4 Efficient Analytical Characterization Of (5.13) *And* (5.18) *For The Fused Lasso*

The fused lasso problem (5.5) can be reformulated as the regression problem

$$\underset{\beta \in \mathbb{R}^T}{\text{minimize}} \left\{ ||y - \boldsymbol{X}\beta||_2^2 + \lambda ||\beta||_1 \right\}, \tag{C.23}$$

for a $T \times T$ matrix $\boldsymbol{X}$ whose jth row contains j ones followed by $T - j$ zeros. (5.5) and (C.23) are equivalent in the sense that $\hat{\beta}_t = \hat{\mu}_t - \hat{\mu}_{t-1}$ for $t = 2, \ldots, T$ and $\hat{\beta}_1 = \hat{\mu}_1$.

Lee et al. (2016) show that the set of y for which the lasso (C.23) results in a given set of selected variables and signs can be written as the polyhedral set $\{y : \boldsymbol{A}y \leq b\}$ for a $T \times T$ matrix $\boldsymbol{A}$ and a T-vector b. $\boldsymbol{A}$ and b have explicit formulas depending only on the selected

variables and coefficient signs. Therefore, Lee et al. (2016) are able to compute p-values for the null hypothesis that the estimated coefficients are zero conditional on the selected variables, the signs of the estimated coefficients, and nuisance parameters.

To avoid conditioning on the signs of the estimated coefficients, we slightly modify the arguments outlined in Section 5.4. In the following propositions, Propositions C.3 and C.4, we modify Propositions 5.2 and 5.3 for $\mathcal{S} = \{\phi : \text{supp}(\hat{\beta}(y'(\phi))) = \text{supp}(\hat{\beta}(y))\}$, where $\text{supp}(\hat{\beta}(y))$ denotes the set of selected variables obtained from solving (C.23) with data y.

Proposition C.3 *The set* $\{\phi : \text{supp}(\hat{\beta}(y'(\phi))) = m, \text{sign}(\hat{\beta}(y'(\phi))) = d\}$ *is an interval. Furthermore, the set* $\mathcal{S} = \{\phi : \text{supp}(\hat{\beta}(y'(\phi))) = \text{supp}(\hat{\beta}(y))\}$ *can be written as the union of intervals,*

$$\mathcal{S} = \{\phi : \text{supp}(\hat{\beta}(y'(\phi))) = \text{supp}(\hat{\beta}(y))\} = \bigcup_{i=-N}^{N'} (a_i, a_{i+1}), \tag{C.24}$$

where $N' + N + 1$ *is the number of elements in the set*

$$\mathcal{I} := \left\{ d : \exists \alpha \in \mathbb{R} \text{ such that } d = \text{sign}(\hat{\beta}(y'(\alpha))), \text{supp}(\hat{\beta}(y)) = \text{supp}(\hat{\beta}(y'(\alpha))) \right\}. \tag{C.25}$$

$\mathcal{I}$ *is the set of possible coefficient signs that can be obtained via a perturbation of* y *that yields the same non-zero coefficients as* $\hat{\beta}(y)$.

Proposition C.4 $\bigcup_{i=-N}^{N'}(a_i, a_{i+1})$ *defined in* (C.24) *can be efficiently computed.*

Now, we outline the proof for Proposition C.4. We first solve (C.23) on the data y in order to obtain $\text{supp}(\hat{\beta}(y))$ and $\text{sign}(\hat{\beta}(y))$. We then apply the first statement in Proposition C.3 to obtain an interval $[a_0, a_1] \subset \mathcal{S}$. Next, for some small positive value of η, we apply the first statement of Proposition C.3 with $m = \text{supp}(\hat{\beta}(y'(a_1 + \eta)))$ and $d = \text{sign}(\hat{\beta}(y'(a_1 + \eta)))$ to identify the interval $[a_1, a_2]$. We then check whether $\text{supp}(\hat{\beta}(y)) = \text{supp}(\hat{\beta}(y'(a_1 + \eta)))$; if so, then $[a_1, a_2] \subset \mathcal{S}$, and if not, then $[a_1, a_2] \not\subset \mathcal{S}$. We continue in this vein, much as we did in Section 5.4, to obtain the full set $\mathcal{S}$.

Generalized Lasso

In this section, we show that we can use the tools from Section 5.4 to characterize the selection event of the generalized lasso. In Section C.4 we rewrote the fused lasso problem (5.5) in terms of a lasso (regression) problem (C.23), which allowed us to develop a simple procedure to characterize $\mathcal{S}$. The generalized lasso (Tibshirani et al., 2011) is the solution to the optimization problem

$$\underset{\beta \in \mathbb{R}^T}{\text{minimize}} \left\{ ||y - \beta||_2^2 + \lambda ||\boldsymbol{D}\beta||_1 \right\}, \tag{C.26}$$

where $\boldsymbol{D}$ is a matrix whose rows encode our beliefs about the underlying structure in the data. For general $\boldsymbol{D}$, (C.26) cannot be rewritten in the form of (C.23), and so existing machinery for selective inference for the lasso cannot be applied. Nonetheless, by also conditioning on the order that variables enter the model, Hyun et al. (2016) show that the selection event of the generalized lasso is polyhedral. Therefore, an extension of the ideas in Section C.4 could be applied in order to conduct selective inference using a smaller conditioning set.

C.5 Timing Results For Estimating Changepoints And Computing p-values

In this section, we present timing results for estimating changepoints and computing p-values. Figure C.3 displays the running time, computed on a MacBook Pro with a 2.5 GHz Intel Core i7 processor, for estimating changepoints and calculating p-values for Approaches 1–4 defined in Section 5.6.1. We take $\lambda = \log(T)$ for ℓ_0 segmentation and use $\max(\hat{K}, 1)-$step binary segmentation for $\hat{K}$ equal to the number of estimated changepoints from ℓ_0 segmentation. Fifty replicate datasets are simulated according to model (5.1) with $\sigma^2 = 1$, and with $K = 10\lfloor \log_{10}(T) \rfloor$ changepoints sampled without replacement from the set $\{1, \ldots, T\}$. At each changepoint, the absolute difference in mean is $|\mu_{\tau_j+1} - \mu_{\tau_j}| = 1.5$. Our implementations of Approaches 1–3 approximate the set $\mathcal{S}$ with $\tilde{\mathcal{S}}$ as described in Proposition 5.4; we take $|a_{-r}| = |a_{r'+1}| = \max(10\sigma||\nu||_2, |\nu^\top y|)$.

Estimating changepoints with binary and ℓ_0 segmentation is very fast (under 0.06 seconds for all series lengths considered). On the other hand, inference is much more costly for all

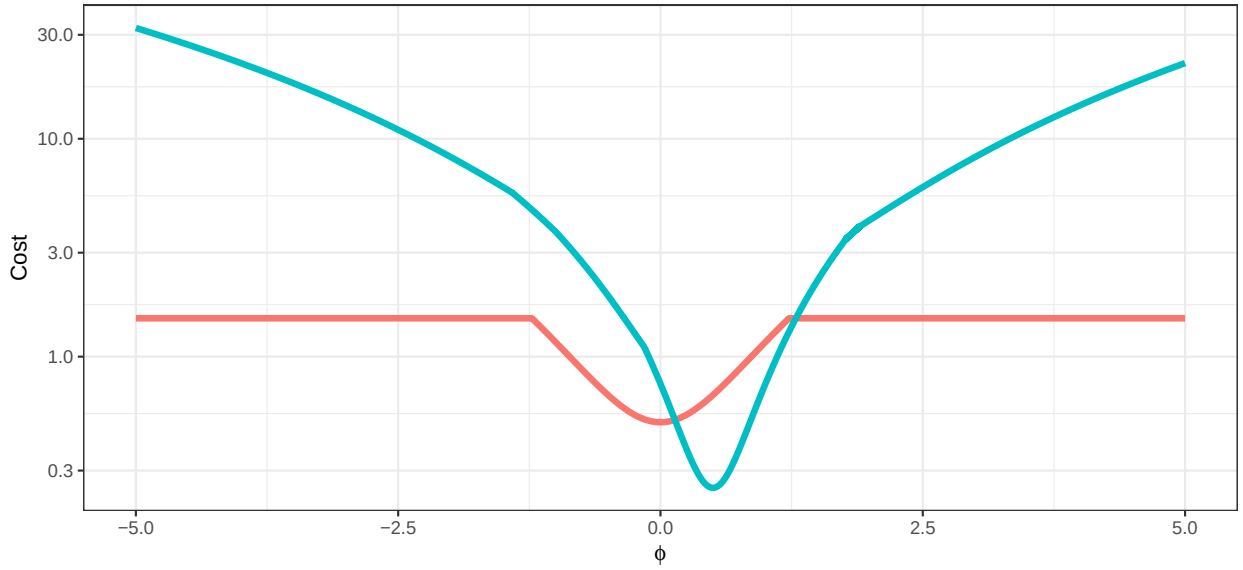


Figure C.1: Optimal cost of segmenting $y'(\phi)$ as a function of ϕ, in the example in Section C.3. $\tilde{C}(\phi)$ is the optimal cost of segmenting $y'(\phi)$ as a function of ϕ given that there is a changepoint at $\hat{\tau} = 3$ (red). $\tilde{C}'(\phi)$ is the optimal cost of segmenting $y'(\phi)$ given that there is no changepoint at $\hat{\tau} = 3$ (blue).

approaches. In particular, we note that Approach 4 is almost an order of magnitude faster than Approaches 1–3 for longer series lengths. We note that Approach 3 can be sped up using the idea presented in Section C.2.

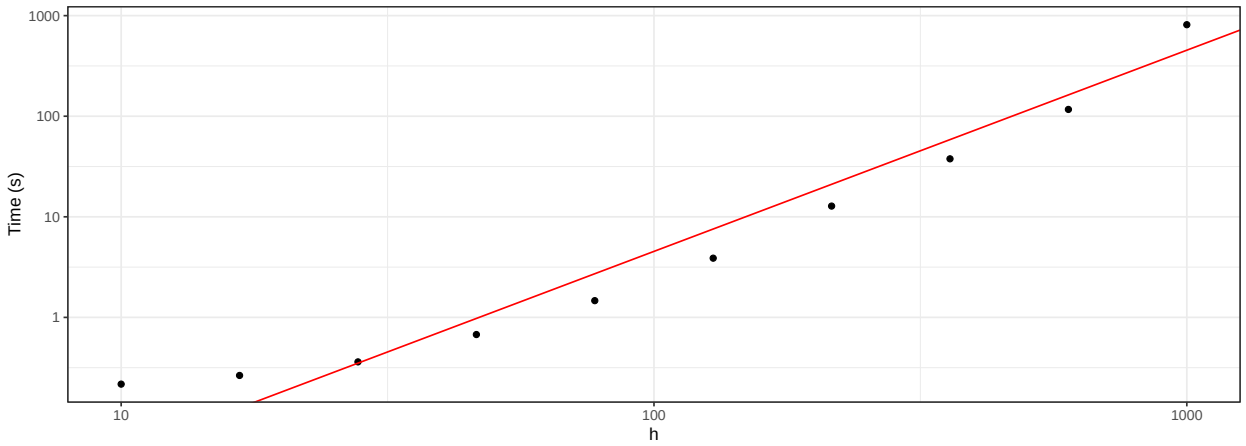


Figure C.2: Average time, in seconds, to compute the set $\mathcal{S}$ in (5.18), as a function of the window size h on 50 replicated datasets. Both axes are displayed on the log scale. The function time $= e^{-3.3}h^2$ (red) is displayed for reference. Details are provided in Appendix C.3.

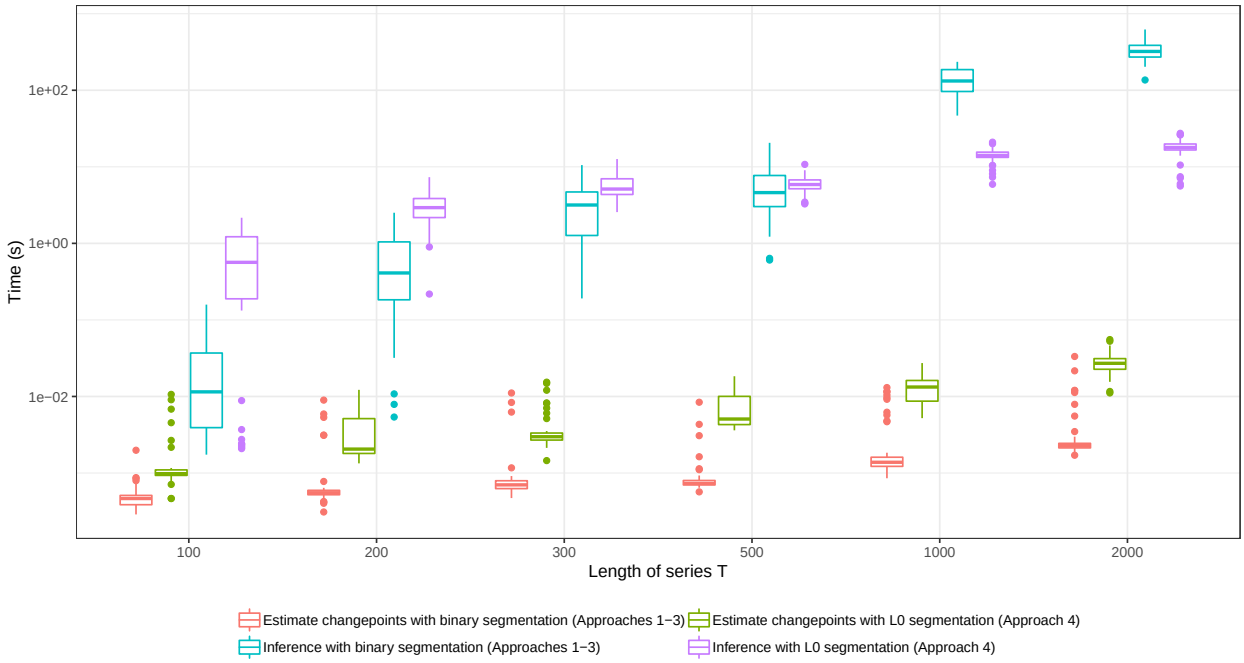


Figure C.3: Computational cost of Approaches 1–4 defined in Section 5.6.1. 50 replicate datasets are simulated according to model (5.1) with $\sigma^2 = 1$ and with $K = 10\lfloor \log_{10}(T) \rfloor$ changepoints sampled without replacement from $\{1, \ldots, T\}$. At each changepoint the absolute difference in mean, $|\mu_{\tau_j+1} - \mu_{\tau_j}|$, is 1.5. Details are provided in Appendix C.5.