Published in final edited form as:

J Comput Graph Stat. 2019 ; 28(4): 954–966. doi:10.1080/10618600.2019.1592758.

Flexible and Interpretable Models for Survival Data

Jiacheng Wu^a, Daniela Witten^{a,b}

^aDepartment of Biostatistics, University of Washington, Seattle, WA;

^bDepartment of Statistics, University of Washington, Seattle, WA

Abstract

As datasets continue to increase in size, there is growing interest in methods for prediction that are both Received January 2018 flexible and interpretable. A flurry of recent work on this topic has focused on additive modeling in the Revised February 2019 regression setting, and in particular, on the use of data-adaptive nonlinear functions that can be used to flexibly model each covariate's effect, conditional on the other features in the model. In this article, we extend this recent line of work to the survival setting. We develop an additive Cox proportional hazards model, in which each additive function is obtained by trend filtering, so that the fitted functions are piece-wise polynomial with adaptively chosen knots. An efficient proximal gradient descent algorithm is used to fit the model. We demonstrate its performance in simulations and in application to a primary biliary cirrhosis data set, as well as a dataset consisting of time to publication for clinical trials in the biomedical literature. Supplementary materials for this article are available online.

Keywords

Additive model; Cox's model; Piece-wise polynomial; Trend filtering

1. Introduction

As datasets continue to increase in size, there is growing interest in the development of statistical models that are simultaneously flexible and interpretable. A number of recent papers have focused on developing flexible and interpretable regression models (Ravikumar et al. 2009; Petersen, Witten, and Simon 2016; Lou et al. 2016; Sadhanala and Tibshirani 2017; Petersen and Witten 2019) within the generalized additive modeling framework (Hastie and Tibshirani 1990a). There has been a particular focus on developing methods that are *data-adaptive*, in the sense that the fitted functions can be either very wiggly or very smooth, as needed based on the data.

In biomedical research, we are often faced with a survival-valued response variable, for which the Cox proportional hazards model is a natural choice (Cox 1972). In this article, we

CONTACT Jiacheng Wu, wuj92@uw.edu, Department of Biostatistics, University of Washington, Seattle, WA 98195.

Color versions of one or more of the figures in the article can be found online at www.tandfonline.com/r/JCGS.

Supplementary materials for this article are available online. Please go to www.tandfonline.com/r/JCGS.

Supplementary Materials

The supplementary materials include R code to produce the tables and figures in this article.

consider the task of extending recent developments for the regression setting to the survival setting, by developing a flexible and interpretable approach for modeling nonlinear predictor effects within the Cox model.

In Section 1.1, we survey the recent literature on flexible and interpretable additive regression modeling. In Section 1.2, we briefly mention existing proposals for nonlinear survival analysis using the Cox model.

1.1. Nonparametric Regression for Continuous Data

Suppose that we wish to predict a continuous-valued response Y from a scalar predictor X , on the basis of n iid observation pairs, (x_i, y_i) , $i = 1, \dots, n$. We consider the model

$$y_i = f(x_i) + \epsilon_i, \quad (1)$$

where ϵ_i is a nonzero error term. A number of approaches are available for estimating the unknown function f in Equation (1); these include kernel regression (Nadaraya 1964; Watson 1964), regression splines (Friedman 1991), smoothing splines (Wahba 1990), wavelets (Daubechies 1992), regression pursuit (Huang, Wei, and Ma 2012), automatic structure discovery (Wu and Stefanski 2015; Zhang, Cheng, and Liu 2011), and others.

Trend filtering is a recent proposal for fitting the model (1), in which it is assumed that f is a piece-wise polynomial, and the knots are not pre-specified (Kim et al. 2009; Tibshirani 2014). Without loss of generality, suppose that $x_1 < \dots < x_n$. Then, trend filtering solves the optimization problem

$$\underset{\theta_1, \dots, \theta_n}{\text{minimize}} \left\{ \sum_{i=1}^n (y_i - \theta_i)^2 + \lambda \|\mathbf{D}^{(X, k+1)} \boldsymbol{\theta}\|_1 \right\}, \quad (2)$$

where λ is a nonnegative tuning parameter, and $\mathbf{D}^{(X, k+1)} \in \mathbb{R}^{(n-k-1) \times n}$ is the discrete difference operator of order $k+1$ with respect to the feature $X = (x_1, \dots, x_n)$. When $k=0$,

$$\mathbf{D}^{(X, 1)} \equiv \begin{pmatrix} 1 & -1 & 0 & \dots & 0 & 0 \\ 0 & 1 & -1 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 1 & -1 \end{pmatrix} \in \mathbb{R}^{(n-1) \times n} \quad (3)$$

is the first-order difference matrix, and therefore $\|\mathbf{D}^{(X, 1)} \boldsymbol{\theta}\|_1 = \sum_{i=2}^n |\theta_i - \theta_{i-1}|$ is the sum of all pairwise differences; in this case, Equation (2) is the one-dimensional fused lasso problem (Tibshirani et al. 2005), which encourages $|\theta_i - \theta_{i-1}|$ to equal zero. Consequently, if we let $\hat{\theta}_1, \dots, \hat{\theta}_n$ denote the solution to Equation (2), and $\hat{f}(x_i) = \hat{\theta}_i$, then $\hat{f}(x_1), \dots, \hat{f}(x_n)$ is piece-wise constant, with adaptively chosen knots. For $k \geq 1$, we define

$$\mathbf{D}^{(X, k+1)} \equiv \tilde{\mathbf{D}}_k^{(X, 1)} \cdot \text{diag}\left(\frac{k}{x_{k+1} - x_1}, \dots, \frac{k}{x_n - x_{n-k}}\right) \cdot \mathbf{D}^{(X, k)} \in \mathbb{R}^{(n-k-1) \times n} \quad (4)$$

recursively, where $\widetilde{\mathbf{D}}_k^{(X,1)} \in \mathbb{R}^{(n-k-1) \times (n-k)}$ in Equation (4) is obtained by omitting the last k columns and k rows from $\mathbf{D}^{(X,1)}$ in Equation (3) (Sadhanala and Tibshirani 2017). Thus, $\mathbf{D}^{(X,k+1)}$ is the discrete analogue of the $(k+1)$ th-order derivative, and solving Equation (2) yields a k th-order piece-wise polynomial with adaptively chosen knots. Trend filtering adapts to the local level of smoothness much better than competing approaches, such as smoothing splines and regression splines (Tibshirani 2014; Sadhanala and Tibshirani 2017).

Now, suppose that we wish to predict Y on the basis of p features, $X_1, \dots, X_p$; we have n iid pairs $(y_i, x_{i1}, \dots, x_{ip})$, $i = 1, \dots, n$. An additive model (Hastie and Tibshirani 1990a; Wood 2006) models the response as

$$y_i = \theta_0 + \sum_{j=1}^p f_j(x_{ij}) + \epsilon_i, \quad (5)$$

where f_j is an unknown mean-zero function associated with the j th covariate. A flurry of recent work has focused on fitting additive models in the high-dimensional setting, with $p > n$ (Lin and Zhang 2006; Meier, Van de Geer, and Bühlmann 2009; Ravikumar et al. 2009; Lou et al. 2016), using spline-based approaches.

However, as shown by Sadhanala and Tibshirani (2017) and Petersen, Witten, and Simon (2016), if the model (5) is fit using linear smoothers such as splines, then it may be unable to adapt to the level of smoothness of the underlying functions. To overcome this problem, the authors proposed fitting the model (5) using trend filtering. Let $\theta_j = (\theta_{1j}, \dots, \theta_{nj})^T \in \mathbb{R}^n$ be the function for the j th covariate in the sense that $\theta_{ij} = f_j(x_{ij})$, and let $\theta_0 \in \mathbb{R}$ be an intercept. Let $\mathbf{P}_j$ be the permutation matrix that orders the elements $x_{1j}, \dots, x_{nj}$ from least to greatest. Additive trend filtering solves the optimization problem

$$\begin{aligned} & \underset{\theta_0, \theta_1, \dots, \theta_p}{\text{minimize}} \left\{ \left\| \mathbf{y} - \sum_{j=1}^p \theta_j - \theta_0 \mathbf{1} \right\|_2^2 + \lambda \sum_{j=1}^p \left\| \mathbf{D}^{(\mathbf{P}_j \mathbf{X}_j, k+1)} \mathbf{P}_j \theta_j \right\|_1 \right\} \text{subject to } \mathbf{1}^T \theta_j \\ & = 0 \text{ for } j = 1, \dots, p. \end{aligned} \quad (6)$$

(Petersen, Witten, and Simon (2016) focused on the case $k=0$, whereas Sadhanala and Tibshirani (2017) allow for $k>1$.) Letting $\hat{\theta}_0, \hat{\theta}_1, \dots, \hat{\theta}_p$ denote the solution to Equation (6), the prediction for the i th observation takes the form $\hat{y}_i = \hat{\theta}_0 + \sum_{j=1}^p \hat{\theta}_{ij} = \hat{\theta}_0 + \sum_{j=1}^p \hat{f}_j(x_{ij})$. The form of each fitted function depends on the value of k ; for example, when $k=1$, $\hat{f}_j(x_{1j}), \dots, \hat{f}_j(x_{nj})$ are piece-wise linear, with adaptively chosen knots. Furthermore, this approach can adapt to the level of smoothness in the underlying function.

1.2. Flexible and Interpretable Modeling of Survival Data

Thus far, the discussion has focused on the setting in which the response variable, Y , is continuous-valued. However, in the case of a (possibly censored) survival response, it is

natural to fit the Cox proportional hazards model. Letting $\mathbf{X}^i = (x_{i1}, \dots, x_{ip})$, we extend the additive modeling framework to this setting,

$$\lambda(t | \mathbf{X}^i) = \lambda_0(t) \exp \left(\sum_{j=1}^p f_j(x_{ij}) \right), \quad (7)$$

where $\lambda(t | \mathbf{X}^i) = \lim_{dt \rightarrow 0} \frac{\Pr(t \leq Y \leq t + dt | Y \geq t, \mathbf{X}^i)}{dt}$ is the conditional hazard given observation

$\mathbf{X}^i$, $\lambda_0(t)$ is the baseline hazard, and f_j is the unknown functional form for the j th covariate. The unknown functions $f_1, \dots, f_p$ can be estimated, for instance, using smoothing splines (O'Sullivan 1988; Hastie and Tibshirani 1990a, 1990b; Gray 1992) or regression splines (Sleeper and Harrington 1990; Kooperberg, Stone, and Truong 1995; Molinari et al. 2001). However, just as in the regression setting, applying such methods requires a researcher to prespecify the level of flexibility—either the degrees of freedom (for smoothing splines) or the number of knots (for regression splines).

1.3. Contribution of This Article

In this article, we propose fitting the additive Cox proportional hazards model (7) using a trend filtering approach (Kim et al. 2009; Tibshirani 2014), so that the fitted functions are piece-wise polynomial with adaptively chosen knots. This approach balances model flexibility and interpretability, and can accommodate the high-dimensional setting, in which the number of covariates is larger than the number of observations.

The rest of this article is organized as follows. In Section 2, we present an approach for trend filtering in the Cox model with a single feature; this generalizes the proposal of Chaturvedi, de Menezes, and Goeman (2014) to the case of higher-order trend filtering. In Section 3, we extend this approach to allow for multiple features. In Section 4, we conduct comprehensive simulation studies to illustrate that the proposed method has better performance than competing methods. In Section 5, we apply the method to a primary biliary cirrhosis dataset from the Mayo Clinic, as well as a dataset consisting of time to publication for clinical trials.

2. Univariate Trend Filtering for the Cox Model

2.1. Review of the Cox Model

For the i th subject, survival data consists of a failure or censoring time t_i and a censoring indicator δ_i , which equals 0 if the i th subject was censored and 1 otherwise. Let $\tau_1 < \tau_2 < \dots < \tau_D$ be the distinct failure times, and for $i = 1, \dots, D$, let $E_i \equiv \{s: t_s = \tau_i, \delta_s = 1\}$ index the subjects who failed at time τ_i . Define $d_i \equiv |E_i|$ as the number of failure events at τ_i . Let $R_i \equiv \{k: t_k \geq \tau_i\}$ be the risk set at time τ_i . Furthermore, let $\mathbf{X} = (x_1, \dots, x_n)$ be a single feature. We assume that the Cox proportional hazards model holds, that is, $\lambda(t | x_i) = \lambda_0(t) \exp(f(x_i))$ for $i = 1, \dots, n$, where $\lambda_0(t)$ is an unspecified baseline hazard. Our goal is to estimate the function f . Define $\theta_i \equiv f(x_i)$ and $\boldsymbol{\theta} = (\theta_1, \dots, \theta_n)^T$. To deal with tie events, we use the Breslow approximation of the partial likelihood (Breslow 1974):

$$L(\boldsymbol{\theta}) = \prod_{i=1}^D \frac{\exp\left(\sum_{s \in E_i} \theta_s\right)}{\left(\sum_{k \in R_i} \exp(\theta_k)\right)^{d_i}}. \quad (8)$$

We define the loss function as the negative log partial likelihood divided by n ,

$$\begin{aligned} l(\boldsymbol{\theta}) &= -\frac{1}{n} \log L(\boldsymbol{\theta}) \\ &= -\frac{1}{n} \sum_{i=1}^D \left[\sum_{s \in E_i} \theta_s - d_i \log \left(\sum_{k \in R_i} \exp(\theta_k) \right) \right]. \end{aligned} \quad (9)$$

2.2. Optimization Problem and Algorithm

Our goal is to flexibly model covariate effects in the Cox model as piece-wise polynomials with adaptively chosen knots. Let $\mathbf{P}$ be the permutation vector that orders the elements of $\mathbf{X}$ from least to greatest. We will consider the optimization problem

$$\underset{\boldsymbol{\theta}}{\text{minimize}} \left\{ l(\boldsymbol{\theta}) + \lambda \|\mathbf{D}^{(\mathbf{P}\mathbf{X}, k+1)} \boldsymbol{\theta}\|_1 \right\} \text{ subject to } \mathbf{1}^T \boldsymbol{\theta} = 0, \quad (10)$$

where $\lambda \geq 0$ is a nonnegative tuning parameter, and $\mathbf{D}^{(\mathbf{P}\mathbf{X}, k+1)} \in \mathbb{R}^{(n-k-1) \times n}$ is the discrete difference operator of order $k+1$ defined in Equations (3) and (4). The constraint $\mathbf{1}^T \boldsymbol{\theta} = 0$ is needed for identifiability.

To solve Equation (10) with $\lambda = 0$, we can use gradient descent, which amounts to repeatedly applying updates of the form

$$\boldsymbol{\theta}^{(h+1)} \leftarrow \boldsymbol{\theta}^{(h)} - s \left(\nabla l(\boldsymbol{\theta}^{(h)}) \right), \quad (11)$$

where h indexes the iterations, $\nabla l(\boldsymbol{\theta})$ is the gradient of the loss function defined in Equation (9), and s is a stepsize chosen to satisfy a condition involving the Lipschitz constant of $\nabla l(\boldsymbol{\theta})$ (Boyd and Vandenberghe 2004) (see Appendix A for details). The gradient has the form

$$(\nabla l(\boldsymbol{\theta}))_u = \frac{1}{n} \sum_{\{i: t_i \leq t_u, \delta_i = 1\}} \frac{d_i \exp(\theta_u)}{\sum_{k \in R_i} \exp(\theta_k)} - \frac{1}{n} \mathbb{1}\{\delta_u = 1\}. \quad (12)$$

for subjects $u = 1, \dots, n$.

We now consider solving Equation (10) for $\lambda > 0$. Since the objective function has the form “convex differentiable” + “convex nondifferentiable,” proximal gradient descent (Beck and Teboulle 2009; Parikh and Boyd 2014) can be used to solve Equation (10). The proximal operator is defined as

$$\text{Prox}_{s\lambda}(\tilde{\boldsymbol{\theta}}) \equiv \underset{\boldsymbol{\theta}}{\text{argmin}} \left\{ \frac{1}{2} \|\tilde{\boldsymbol{\theta}} - \boldsymbol{\theta}\|_2^2 + s\lambda \|\mathbf{D}^{(\mathbf{P}\mathbf{X}, k+1)} \boldsymbol{\theta}\|_1 \right\}, \quad (13)$$

and can be easily computed by applying existing solvers for trend filtering, such as the R package `glmgen` (Arnold, Sadhanala, and Tibshirani 2014). In the $(h+1)$ th iteration, the proximal gradient update takes the form

$$\boldsymbol{\theta}^{(h+1)} \leftarrow \text{Prox}_{s\lambda}(\boldsymbol{\theta}^{(h)} - s(\nabla l(\boldsymbol{\theta}^{(h)}))). \quad (14)$$

3. Additive Trend Filtering for the Cox Model

3.1. Optimization Problem

Now suppose we have p covariates, $\mathbf{X}_1, \dots, \mathbf{X}_p \in \mathbb{R}^n$. Our goal is to fit a model of the form Equation (7), where $f_1, \dots, f_p$ are piece-wise polynomial, with adaptively chosen knots. Define $\theta_{ij} = f_j(x_{ij})$, so that the model of interest takes the form $\lambda(t | \mathbf{X}_i) = \lambda_0(t) \exp(\sum_{j=1}^p \theta_{ij})$, and let $\boldsymbol{\theta} = (\boldsymbol{\theta}_1^T, \dots, \boldsymbol{\theta}_p^T)^T \in \mathbb{R}^{np}$. Then the partial likelihood can be expressed as

$$L(\boldsymbol{\theta}) = \prod_{i=1}^D \frac{\exp(\sum_{s \in E_i} \sum_{j=1}^p \theta_{sj})}{(\sum_{k \in R_i} \exp(\sum_{j=1}^p \theta_{kj}))^{d_i}}, \quad (15)$$

leading to the loss function

$$\begin{aligned} l(\boldsymbol{\theta}) &= -\frac{1}{n} \log L(\boldsymbol{\theta}) \\ &= -\frac{1}{n} \sum_{i=1}^D \left[\sum_{s \in E_i} \sum_{j=1}^p \theta_{sj} - d_i \log \left(\sum_{k \in R_i} \exp \left(\sum_{j=1}^p \theta_{kj} \right) \right) \right]. \end{aligned} \quad (16)$$

The gradient for the u th subject and the v th feature takes the form

$$(\nabla l(\boldsymbol{\theta}))_{uv} = \frac{1}{n} \sum_{\{i: t_i \leq t_u, \delta_i = 1\}} \frac{d_i \exp(\sum_{j=1}^p \theta_{uj})}{\sum_{k \in R_j} \exp(\sum_{j=1}^p \theta_{kj})} - \frac{1}{n} \mathbb{I}\{\delta_u = 1\}, \quad (17)$$

where $u = 1, \dots, n$ and $v = 1, \dots, p$.

Let $\mathbf{P}_j$ be the permutation matrix that orders the elements of $\mathbf{X}_j$ from least to greatest. As in Petersen, Witten, and Simon (2016), we will apply two penalties to Equation (16): a trend filtering penalty to encourage λ to be piece-wise polynomial, and an ℓ_2 or group lasso penalty to encourage a coefficient to be zeroed out completely (Yuan and Lin 2006; Simon and Tibshirani 2012). The optimization problem is

$$\underset{\boldsymbol{\theta}}{\text{minimize}} \left\{ l(\boldsymbol{\theta}) + \alpha \lambda \sum_{j=1}^p \|\mathbf{D}^{(\mathbf{P}_j \mathbf{X}_j, k+1)} \mathbf{P}_j \boldsymbol{\theta}_j\|_1 + (1-\alpha) \lambda \sum_{j=1}^p \|\boldsymbol{\theta}_j\|_2 \right\}. \quad (18)$$

In Equation (18), the parameter α controls the tradeoff between these two penalties. We will refer to Equation (18) as the optimization problem for the trend filtering Cox model; it is the main focus of this article.

Proposition 1 indicates that constraints of the form $\mathbf{1}^T \boldsymbol{\theta}_j = 0$ for $j = 1, \dots, p$ are not needed in Equation (18). It is proven in Appendix B.

Proposition 1. For $\lambda > 0$ and $\alpha < 1$, let $\hat{\boldsymbol{\theta}}_1, \dots, \hat{\boldsymbol{\theta}}_p$ denote the solution to Equation (18). Then $\mathbf{1}^T \hat{\boldsymbol{\theta}}_j = 0$ for $j = 1, \dots, p$.

3.2. Algorithm

We will use proximal gradient descent to solve Equation (18). Similar to Equation (13), the proximal operator of Equation (18) for the j th feature is defined as

$$\text{Prox}_{s\lambda}(\tilde{\boldsymbol{\theta}}_j) \equiv \underset{\boldsymbol{\theta}_j}{\operatorname{argmin}} \left\{ \frac{1}{2} \|\tilde{\boldsymbol{\theta}}_j - \boldsymbol{\theta}_j\|_2^2 + s\lambda\alpha \left\| \mathbf{D}^{(\mathbf{P}_j \mathbf{X}_{j,k+1})} \mathbf{P}_j \boldsymbol{\theta}_j \right\|_1 + s\lambda(1-\alpha) \|\boldsymbol{\theta}_j\|_2 \right\} \quad (19)$$

However, due to the presence of the group lasso penalty, Equation (19) is not a standard trend filtering problem, and it is not obvious how to solve it. The following proposition follows directly from Lemma 7.1 of Petersen, Witten, and Simon (2016).

Proposition 2. Define

$$\hat{\boldsymbol{\theta}}_j = \underset{\boldsymbol{\theta}_j}{\operatorname{argmin}} \left\{ \frac{1}{2} \|\tilde{\boldsymbol{\theta}}_j - \boldsymbol{\theta}_j\|_2^2 + s\lambda\alpha \left\| \mathbf{D}^{(\mathbf{P}_j \mathbf{X}_{j,k+1})} \mathbf{P}_j \boldsymbol{\theta}_j \right\|_1 \right\}, \quad (20)$$

and

$$\bar{\boldsymbol{\theta}}_j = \underset{\boldsymbol{\theta}_j}{\operatorname{argmin}} \left\{ \frac{1}{2} \|\tilde{\boldsymbol{\theta}}_j - \boldsymbol{\theta}_j\|_2^2 + s\lambda\alpha \left\| \mathbf{D}^{(\mathbf{P}_j \mathbf{X}_{j,k+1})} \mathbf{P}_j \boldsymbol{\theta}_j \right\|_1 + s\lambda(1-\alpha) \|\boldsymbol{\theta}_j\|_2 \right\}. \quad (21)$$

Then,

$$\bar{\boldsymbol{\theta}}_j = \left(1 - \frac{s\lambda(1-\alpha)}{\|\hat{\boldsymbol{\theta}}_j\|_2} \right)_+ \hat{\boldsymbol{\theta}}_j. \quad (22)$$

Proposition 2 indicates that rather than directly solving Equation (19), we can instead solve a standard trend filtering problem, and then multiply the solution by a scalar to obtain the solution to Equation (19).

The proximal gradient descent algorithm is detailed in Algorithm 1. While a fixed step size could be chosen based on the Lipschitz constant of $\nabla \ell(\boldsymbol{\theta})$ (see Appendix A), we instead use backtracking line search (Beck and Teboulle 2009). Convergence is linear in the number of iterations, in the sense that if we let $g(\cdot)$ denote the objective of Equation (18), then $g(\boldsymbol{\theta}^{(k)}) - g(\boldsymbol{\theta}^*) \leq C \|\boldsymbol{\theta}^{(k)} - \boldsymbol{\theta}^*\|^2 / k$ for some constant C , where $\boldsymbol{\theta}^{(k)}$ is the value of

$\theta = (\theta_1^T, \dots, \theta_p^T)^T$ in the k th iteration of proximal gradient descent and $\theta^* = \underset{\theta}{\operatorname{argmin}} g(\theta)$ (Beck and Teboulle 2009).

We typically solve Equation (18) with a grid of λ values, for a fixed value of α , using an efficient “warm start” procedure. In particular, we begin by applying Algorithm 1 with a very large value of λ , which we call $\lambda_{\max}$, for which the solution to Equation (18) is easy to compute; details are provided in Appendix C. Next, we solve Algorithm 1 using a slightly smaller value of λ , initializing $\theta^{(0)}$ using the previous solution obtained. This process is repeated for a fine grid of decreasing λ values that are equally spaced on the log scale between $\lambda_{\max}$ and $0.01\lambda_{\max}$.

The computation time of Algorithm 1 depends mainly on the computation of the gradient, which requires $O(n^2 + np)$ computations per iteration, and the computation of the proximal operator, which takes $O(np)$ computations per iteration (this is the computation time of univariate trend filtering using ADMM; see Ramdas and Tibshirani 2016). Therefore, the overall computational complexity for a fixed pair (α, λ) is $O(n^2 + np)$ per iteration. Timing results are presented in Table 1 in Section 4.

Algorithm 1

Proximal Gradient Descent for Cox Trend Filtering (18)

-
1. Initialize $\theta_j^{(0)} \leftarrow \mathbf{0}$, and set $\theta_j^{\text{new}} \leftarrow \theta_j^{(0)}$, for $j = 1, \dots, p$.
 2. Set $s = 25$ and $\beta = 0.8$.
 3. For $h = 0, 1, 2, \dots$ until $\|\theta^{(h+1)} - \theta^{(h)}\|^2 \leq 10^{-6}$:
 - (a) While $l(\theta^{\text{new}}) \geq l(\theta^{(h)}) + (\nabla l(\theta^{(h)}))^T(\theta^{\text{new}} - \theta^{(h)}) + \frac{1}{2s} \|\theta^{\text{new}} - \theta^{(h)}\|_2^2$:
 - i. Set $s \leftarrow \beta s$.
 - ii. For each covariate $j = 1, \dots, p$:

- A. Take a gradient step,

$$\tilde{\theta}_j \leftarrow \theta_j^{(h)} - s \left(\nabla l(\theta^{(h)}) \right)_j.$$

- B. Use an existing solver, such as the `glmgen` package in R (Arnold, Sadhanala, and Tibshirani 2014), to compute

$$\theta_j^{\text{new}} \leftarrow \underset{\theta_j}{\operatorname{argmin}} \left\{ \frac{1}{2} \|\tilde{\theta}_j - \theta_j\|_2^2 + s\lambda\alpha \left\| \mathbf{D}^{(\mathbf{P}_j \mathbf{X}_j, k+1)} \mathbf{P}_j \theta_j \right\|_1 \right\}.$$

- C. Soft-scale the estimate,

$$\theta_j^{\text{new}} \leftarrow \left(1 - \frac{(1 - \alpha)s\lambda}{\|\theta_j^{\text{new}}\|_2} \right)_+ \theta_j^{\text{new}}.$$

- (b) Set $\theta^{(h+1)} \leftarrow \theta^{\text{new}}$.

3.3. Incorporating Discrete Covariates Into Equation (18)

Thus far, we have only considered continuous-valued covariates in the Cox trend filtering problem. However, discrete-valued covariates occur frequently in survival analysis, and we may not want to penalize their coefficients. Let $A \subset \{1, \dots, p\}$ be the set of continuous covariates and $\mathcal{A}^c$ be the set of discrete covariates.

Then Equation (18) generalizes

$$\begin{aligned} & \underset{\theta}{\text{minimize}} \left\{ l(\theta) + \alpha \lambda \sum_{j \in \mathcal{A}} \left\| \mathbf{D}^{(\mathbf{P}_j \mathbf{X}_{j \cdot}^{k+1})} \mathbf{P}_j \theta_j \right\|_1 + (1 - \alpha) \lambda \sum_{j \in \mathcal{A}^c} \|\theta_j\|_2 \right. \\ & \left. \right\} \text{subject to } \theta_{ij} = \theta_{uj} \text{ if } X_{ij} = X_{uj} \text{ for } j \in \mathcal{A}^c, \end{aligned} \quad (23)$$

where the constraints, $\theta_{ij} = \theta_{uj}$ if $X_{ij} = X_{uj}$, force covariates corresponding to a common value of the discrete-valued feature to have the same coefficients. Algorithm 2 in Appendix D solves Equation (23).

4. Simulation Study

Our simulation setting is motivated by that of Petersen, Witten, and Simon (2016). Survival data is generated from the Cox model with Exponential(1) baseline hazard, so that $\lambda_0(t) = 1$ and

$$\lambda(t | \mathbf{X}_i) = \exp \left(\sum_{j=1}^p f_j(x_{ij}) \right), \quad (24)$$

where x_{ij} is iid Uniform $[-2.5, 2.5]$. We generated four signal functions, $f_1, \dots, f_4$, according to four scenarios:

Scenario 1: Piece-wise constant functions

Scenario 2: Smooth functions

Scenario 3: Piece-wise linear functions

Scenario 4: Constant in some areas and highly variable in other areas

All functions are standardized such that $\int_{-2.5}^{2.5} f_j = 0$ and $\int_{-2.5}^{2.5} f_j^2 = 1$. In the low-dimensional setting, we have $p = 4$ signal functions, and in the high-dimensional setting, we have 4 signal functions plus 96 noise functions ($f_5, \dots, f_{100} = 0$). Figure 1 shows the functions considered in the four simulation scenarios. We consider a sample size of $n = 100$. We consider uninformative censoring with an Exponential(0.17) distribution, such that approximately 25% of the subjects are censored.

We compare constant Cox trend filtering ($k = 0$ in (18)) and linear Cox trend filtering ($k = 1$ in Equation (18)) to two competing methods: (i) generalized additive models (GAM) using smoothing splines, implemented in the R package *mgcv* (Wood and Wood 2016); and (ii) sparse group lasso (SGL) with natural cubic splines as the basis vectors, and with knots at

the quantile values of the features, implemented in the R package SGL (Simon et al. 2013). For the latter, we fit a model of the form $\lambda(t | \mathbf{X}_i) = \lambda_0(t) \exp \left(\sum_{j=1}^p (\boldsymbol{\beta}^{(j)})^T \mathbf{X}_i^{(j)} \right)$, where $\mathbf{X}_i^{(j)}$ is the natural cubic spline basis vector corresponding to x_{ij} and $\boldsymbol{\beta}^{(j)}$ is the corresponding coefficient vector, by solving the optimization problem

$$-\frac{1}{n} \sum_{i=1}^D \left[\sum_{s \in E_i} \sum_{j=1}^p (\boldsymbol{\beta}^{(j)})^T \mathbf{X}_s^{(j)} - d_i \log \left(\sum_{k \in R_i} \exp \left(\sum_{j=1}^p (\boldsymbol{\beta}^{(j)})^T \mathbf{X}_k^{(j)} \right) \right) \right] + (1 - \alpha) \lambda \sum_{j=1}^p \|\boldsymbol{\beta}^{(j)}\|_2 + \alpha \lambda \sum_{j=1}^p \|\boldsymbol{\beta}^{(j)}\|_1.$$

We generate training, testing, and validation sets, each with sample size 100, from Equation (24) for each of the four scenarios, in both the low- and high-dimensional settings. We fit the models on the training set using a range of tuning parameter values. We perform Cox trend filtering using a fine grid of λ values, with $\alpha = 1$ in the low-dimensional setting, and with $\alpha \in \{0.1, 0.2, \dots, 0.9\}$ in the high-dimensional setting. For GAM, we consider a range of degrees of freedom, $d \in \{4, 5, 6, 7, 8, 9\}$, for the smoothing splines. For SGL, we consider $\alpha = 1$ in the low-dimensional setting, $\alpha \in \{0.1, 0.2, \dots, 0.9\}$ in the high-dimensional setting, and a fine grid of values of λ . We identify the tuning parameter values such that the model fit on the training set minimizes the loss Equation (16) evaluated on the test set, and we refer to the model fit on the training set using these tuning parameter values as the “selected model.” For this selected model, we report the loss on the validation dataset, the parameter fit error $\sum_{j=1}^p \|\theta_j - \hat{\theta}_j\|_2^2$, and the number of knots. Tables 2 and 3 show the simulation results in the low-dimensional and high-dimensional settings, respectively, averaged over 100 simulated datasets. Note that we can apply GAM only in low-dimensions.

In the low-dimensional setting, constant trend filtering outperforms GAM and SGL in scenarios 1 and 4, in which the true underlying functions can be well-approximated by piece-wise constant functions. Linear trend filtering and GAM perform the best in scenarios 2 and 3, in which the true underlying functions are smooth. Linear trend filtering and GAM have similar performance across all simulation scenarios.

In the high-dimensional setting, constant trend filtering performs the best in scenarios 1 and 4, and linear trend filtering performs the best in scenarios 2 and 3. We note that constant trend filtering, linear trend filtering, and sparse group lasso select a large number of truly null features: in particular, they select around 20% of features, whereas in reality, only 4% of features are truly nonzero. We can make sense of this behavior by drawing an analogy to the lasso in the least-squares setting. When prediction accuracy is used to select the tuning parameter for the lasso, the lasso is, in general, not consistent for variable selection (Leng, Lin, and Wahba 2006; Meinshausen and Bühlmann 2006), and many noise variables are selected in the model. Thus, the tuning parameter value selected by cross-validation is good for prediction, but not good for variable selection. Therefore, it is not surprising that when we select the tuning parameter for Cox trend filtering using the aforementioned train-test-validation approach, we tend to select too many features.

To summarize, if the true underlying function is highly variable and exhibits different degrees of smoothness across its domain, then constant trend filtering can approximate it very well. If the true underlying function is smooth, then linear trend filtering has better performance. Figure 2 shows the estimates $\hat{\theta}_1, \hat{\theta}_2, \hat{\theta}_3, \hat{\theta}_4$ from constant trend filtering, linear trend filtering, and GAM in scenario 4, with tuning parameters selected based on the test set. Constant trend filtering can better capture the local smoothness of the underlying functions relative to linear trend filtering and GAM. This explains why constant trend filtering performs the best in this setting.

Table 1 shows the running times of constant and linear trend filtering, GAM, and SGL. Constant trend filtering, GAM, and SGL have similar running times, while linear trend filtering is quite a bit slower; this is reflective of the fact that linear trend filtering is more computationally demanding than constant trend filtering (Ramdas and Tibshirani 2016).

5. Application

5.1. Primary Biliary Cirrhosis Data

In this section, we consider a dataset from a primary biliary cirrhosis (PBC) clinical trial, conducted between 1974 and 1984 in the Mayo Clinic (Fleming and Harrington 1991). The data are available in the R package *survival*. 312 patients participated in the randomized placebo-controlled trial of the drug D-penicillamine during the ten-year period. The primary outcome is time to death and the follow-up time is around 11 years. We use predictive mean matching to impute missing data, using the R package *mice*, as described in Section 3.4 of Van Buuren (2018). We perform the analysis on a single imputed dataset. There are 10 continuous-valued and seven discrete-valued covariates; discrete-valued covariates are handled as in Equation (23). We compare the parameter fits obtained using constant Cox trend filtering, linear Cox trend filtering, GAM (implemented in the R package *mgcv*), and the Cox model with linear predictors (implemented in the R package *survival*).

Figure 3 shows the estimated log hazard ratios for eight out of the ten continuous-valued variables, with tuning parameters chosen by cross-validation. We see that our proposal has the potential to be more interpretable than GAM or the Cox model with linear predictors, since changes in the log hazard ratio occur only at discrete values of the features. For example, in the top left panel, we see that the dashed curve corresponding to constant trend filtering increases by around 0.3 at approximately 50 years of age. Thus, the hazard of death increases by a factor of around 1.35 at 50 years of age.

To evaluate the performance of each method, we also split the data into equally sized training, testing, and validation sets. As in Section 4, we fit models on the training set, selected tuning parameter values on the test set, and reported the loss in Equation (16) and area under the curve (AUC) on the validation set. We use the nearest neighbor estimation method in Heagerty, Lumley, and Pepe (2000), implemented in the R package *survivalROC* to compute AUC on the validation set at the median survival time (around 5.5 years). The results over 100 splits are shown in Table 4. We find that constant and linear trend filtering result in lower loss than GAM and the Cox model with linear predictors, whereas all methods have similar predictive performance in terms of AUC.

5.2. Time to Publication for Clinical Trials

Fast publication of the results from clinical trials is critical for the dissemination of research results as well as the responsible stewardship of taxpayer dollars. Gordon et al. (2013) analyzed 244 extramural randomized cardiovascular clinical trials that were supported by the National Heart, Lung, and Blood Institute. The selected trials were completed between January 1, 2000, and December 31, 2011. The primary outcome was the time between completion of the trial and publication of the main results in a peer-reviewed journal. Similar to the PBC analysis, Figure 4 shows the estimated log hazard ratios for three continuous-valued variables, with tuning parameters chosen by cross-validation. All the methods indicate that studies with larger budgets tend to have shorter time to publication (after adjusting for all other covariates). However, the linear Cox model, linear trend filtering, and GAM indicate that the log hazard increases linearly with the budget. By contrast, the results for constant trend filtering indicate that the log hazard increases by around 0.2 at a \$2.9 million budget, and increases by around 0.4 at a \$5.9 million budget. In other words, given that a study has a budget of at least \$5.9 million, a further increase to the budget is not associated with faster publication time.

We use the approach described in Section 5.1 to evaluate the performance of each method. Table 5 shows the performance of each method on the validation set, averaged over 100 data splits. We find that constant filtering result in lower loss and higher predictive performance in terms of AUC than linear trend filtering, GAM and the Cox model with linear predictors.

6. Discussion

In this article, we exploit recent developments in additive trend filtering to propose an approach for fitting an additive Cox model, in which each fitted function is piece-wise polynomial with adaptively chosen knots. Our proposal adapts to the local smoothness of the underlying functions better than linear smoothers. This agrees with recent results in the linear regression setting (Tibshirani 2014; Petersen, Witten, and Simon 2016; Sadhanala and Tibshirani 2017). In two applications, we show that our proposal balances the need for flexibility and interpretability in modeling nonlinear covariate effects.

In the least-squares setting, there is a well-developed notion of the degrees of freedom of an estimator (Efron 1986; Hastie and Tibshirani 1990a). This makes it possible to compare the complexity of two different estimators. However, to the best of our knowledge, there is no analogous notion of degrees of freedom in the Cox model. Therefore, it is not possible to directly compare the complexity of our estimator to, say, the Cox model with linear predictors. We leave this as a possible avenue for future work.

A number of authors have studied the theoretical properties of the ℓ_1 -penalized Cox model (Huang et al. 2013; Kong and Nan 2014), as well as additive trend filtering (Tibshirani 2014; Sadhanala and Tibshirani 2017). We leave an investigation of the theoretical aspects of our proposal to future work.

This article focuses on estimation, rather than on inference. Inference within the context of regularized regression models is a challenging area of ongoing research, even within the

much simpler least-squares setting (Park and Casella 2008; Zhang and Zhang 2014; Bühlmann 2013; Van de Geer et al. 2014; Javanmard and Montanari 2014b, 2014a; Meir and Drton 2017). A recent line of work on post-selection inference or selective inference (Lockhart et al. 2014; Lee et al. 2016; Tibshirani et al. 2016) could potentially be used to conduct inference conditional on the set of knots selected by Cox trend filtering. This is a difficult problem, since the set of knots is selected based on the data, and so classical statistical inference tools that require the knots to be known a priori will fail. In the much simpler setting of trend filtering with a single covariate and a Gaussian response, Hyun, G'Sell, and Tibshirani (2016) recently proposed a selective inference approach. Generalizing that framework to this setting, in which we have a survival response and multiple covariates, is a challenging direction for future research.

The R package `tfCox` will be made available on CRAN.

Supplementary Material

Refer to Web version on PubMed Central for supplementary material.

Acknowledgments

The authors grateful to Noah Simon and Ashley Petersen for helpful discussions. We are also grateful to three referees, the associate editor, and the editor, for helpful comments and suggestions.

Funding

This work was partially supported by NIH funding: DP5OD009145, NSF CAREER Award DMS-1252624, and a Simons Investigator Award in Mathematical Modeling of Living Systems to Daniela Witten.

Appendix

Appendix A: Lipschitz Continuity of the Gradient of the Loss Function

Define $\theta = (\theta_1^T, \dots, \theta_p^T)^T \in \mathbb{R}^{np}$. The gradient ∇l in Equation (17) is Lipschitz continuous with constant M if

$$\|\nabla l(\theta) - \nabla l(\theta')\|_2 \leq M \|\theta - \theta'\|_2,$$

for any $\theta, \theta' \in \mathbb{R}^{np}$ (Parikh and Boyd 2014). Define

$$\pi_{iu} \equiv \begin{cases} \frac{\exp\left(\sum_{j=1}^p \theta_{uj}\right)}{\sum_{k \in R_i} \exp\left(\sum_{j=1}^p \theta_{kj}\right)} & \text{if } t_i \leq t_u, \\ 0 & \text{otherwise} \end{cases},$$

for i such that $\delta_i = 1$. The Hessian $\nabla^2 l$ has components

$$\begin{aligned}\frac{\partial^2 l}{\partial \theta_{uv}^2} &= \frac{1}{n} \sum_{\{i: t_i \leq t_u, \delta_i = 1\}} d_i (\pi_{iu} - \pi_{iu}^2) \\ &= \frac{1}{n} \sum_{\{i: \delta_i = 1\}} d_i (\pi_{iu} - \pi_{iu}^2),\end{aligned}$$

$$\begin{aligned}\frac{\partial^2 l}{\partial \theta_{uv} \partial \theta_{u'v'}} &= -\frac{1}{n} \sum_{\{i: t_i \leq \min(t_u, t_{u'}), \delta_i = 1\}} d_i \pi_{iu} \pi_{iu'} \\ &= -\frac{1}{n} \sum_{\{i: \delta_i = 1\}} d_i \pi_{iu} \pi_{iu'},\end{aligned}$$

where $u, u' \in \{1, \dots, n\}$, $v, v' \in \{1, \dots, p\}$, and $u \neq u'$. Notice that $\frac{\partial^2 l}{\partial \theta_{uv}^2} = \frac{\partial^2 l}{\partial \theta_{nv'}^2}$ and

$\frac{\partial^2 l}{\partial \theta_{uv} \partial \theta_{u'w}} = \frac{\partial^2 l}{\partial \theta_{u'v} \partial \theta_{uw}}$ for all $v, v', w, w' = 1, \dots, p$. Define $\mathbf{A}$ to be the $n \times n$ matrix for

which the (u, u') th element equals $\frac{\partial^2 l}{\partial \theta_{uv} \partial \theta_{u'v}}$. Therefore, $\nabla^2 l = (\mathbf{1}_p \mathbf{1}_p^T) \otimes \mathbf{A}$.

For any vector $\mathbf{X} = (\mathbf{X}_1^T, \dots, \mathbf{X}_p^T)^T \in \mathbb{R}^{np}$ where $\mathbf{X}_1^T, \dots, \mathbf{X}_p^T \in \mathbb{R}^n$ define $\mathbf{y} \equiv \mathbf{X}_1 + \dots + \mathbf{X}_p$, and consider the quadratic form

$$\begin{aligned}\mathbf{x}^T (\nabla^2 l) \mathbf{x} &= (\mathbf{X}_1^T + \dots + \mathbf{X}_p^T) \mathbf{A} (\mathbf{X}_1 + \dots + \mathbf{X}_p) \\ &= \mathbf{y}^T \mathbf{A} \mathbf{y} \\ &= \sum_{u=1}^n y_u^2 A_{uu} + 2 \sum_{u=1}^n \sum_{l:l > u} y_u y_l A_{ul} \\ &= \sum_{u=1}^n \left[\frac{y_u^2}{n} \sum_{i: \delta_i = 1} d_i (\pi_{iu} - \pi_{iu}^2) \right] - 2 \sum_{u=1}^n \left[\sum_{l:l > u} \frac{y_u y_l}{n} \sum_{i: \delta_i = 1} d_i \pi_{iu} \pi_{il} \right] \\ &= \frac{1}{n} \sum_{u=1}^n \sum_{i: \delta_i = 1} d_i y_u^2 \pi_{iu} - \frac{1}{n} \sum_{u=1}^n \sum_{i: \delta_i = 1} d_i \left(y_u^2 \pi_{iu}^2 + 2 \sum_{l:l > u} y_u y_l \pi_{iu} \pi_{il} \right) \\ &= \frac{1}{n} \sum_{u=1}^n \sum_{i: \delta_i = 1} d_i y_u^2 \pi_{iu} - \frac{1}{n} \sum_{i: \delta_i = 1} d_i \left(\sum_{u=1}^n y_u \pi_{iu} \right)^2 \\ &\leq \frac{1}{n} \sum_{i: \delta_i = 1} \sum_{u=1}^n d_i y_u^2 \pi_{iu} \\ &\leq \frac{1}{n} \sum_{i: \delta_i = 1} d_i \|\mathbf{y}\|_2^2 \\ &\leq \|\mathbf{y}\|_2^2 \\ &\leq \sum_{i=1}^n \|\mathbf{X}_i\|_2^2 \\ &= \|\mathbf{X}\|_2^2.\end{aligned}$$

Since the maximum eigenvalue of $\nabla^2 I$ is $\sup_{\mathbf{X} \neq 0} \frac{\mathbf{X}^T \nabla^2 I \mathbf{X}}{\mathbf{X}^T \mathbf{X}}$, the maximum eigenvalue of $\nabla^2 I$ is bounded above by 1. Therefore, $\nabla^2 I \leq I$ and ∇I is Lipschitz continuous with constant $M = 1$. If the stepsize $s = 1/M$, then proximal gradient descent is guaranteed to converge to the global minimum (Parikh and Boyd 2014).

Appendix

Appendix B: Proof of Proposition 1

Let $(\hat{\theta}_1, \dots, \hat{\theta}_p)$ be the minimum of Equation (18) and $\bar{\theta}_j = \frac{1}{n} \mathbf{1}^T \hat{\theta}_j$ be the mean. Suppose $\hat{\theta}_j$ is not centered, that is, there exists a $v \in \{1, \dots, p\}$ such that $\bar{\theta}_v \neq 0$. Notice that the partial likelihood is invariant to centering, $l(\hat{\theta}_1, \dots, \hat{\theta}_p) = l(\hat{\theta}_1 - \bar{\theta}_1 \mathbf{1}, \dots, \hat{\theta}_p - \bar{\theta}_p \mathbf{1})$, as is the trend filtering penalty, $\|\mathbf{D}(\mathbf{P}_j \mathbf{X}_{j, k+1}) \mathbf{P}_j \hat{\theta}_j\|_1 = \|\mathbf{D}(\mathbf{P}_j \mathbf{X}_{j, k+1}) \mathbf{P}_j (\hat{\theta}_j - \bar{\theta}_j \mathbf{1})\|_1$. But $\|\hat{\theta}_j - \bar{\theta}_j \mathbf{1}\|_2 \leq \|\hat{\theta}_j\|_2$ for $j = 1, \dots, p$, and in particular, $\|\hat{\theta}_v - \bar{\theta}_v \mathbf{1}\|_2 \leq \|\hat{\theta}_v\|_2$. Therefore, centering the vectors $\hat{\theta}_1, \dots, \hat{\theta}_p$ will lead to a smaller value of the objective in Equation (18). This contradicts the assumption that $(\hat{\theta}_1, \dots, \hat{\theta}_p)$ is the minimum of Equation (18). Therefore, $\mathbf{1}^T \hat{\theta}_j = 0$ for all $j = 1, \dots, p$.

Appendix

Appendix C: Derivation of Minimum λ Such That Both Penalty Terms Are Zero

Here, we derive an expression for the value $\lambda_{\max}$ that guarantees that the penalty terms are entirely equal to zero in Equation (18). To simplify notation, write $\mathbf{D}(\mathbf{P}_j \mathbf{X}_{j, k+1})$ as $\mathbf{D}_j$. We consider three cases $\alpha = 1$, $\alpha = 0$, and $0 < \alpha < 1$.

Proposition 3. Let $\tilde{\theta}_1, \dots, \tilde{\theta}_p$ be the solution to

$$\underset{\theta_1, \dots, \theta_p}{\text{minimize}} \ l(\boldsymbol{\theta}), \text{ subject to } \mathbf{D}_j \mathbf{P}_j \boldsymbol{\theta}_j = 0 \text{ for } j = 1, \dots, p. \quad (\text{C.1})$$

If $\alpha = 1$ in Equation (18), then $\lambda = \lambda_{\max, 1}$, where $\lambda_{\max, 1} \equiv \max_j \|(\mathbf{D}_j \mathbf{P}_j^T)^{-1} \mathbf{D}_j \mathbf{P}_j (\nabla l(\tilde{\theta}))_j\|_{\infty}$ is a necessary and sufficient condition for $\tilde{\theta}_1, \dots, \tilde{\theta}_p$ to be the solution to Equation (18).

Proof. To begin, notice that because $\|(\nabla l(\boldsymbol{\theta}))_j\|_{\infty} \leq 2$, it follows that

$$\max_j \sup_{\boldsymbol{\theta}} \|(\mathbf{D}_j \mathbf{P}_j^T)^{-1} \mathbf{D}_j \mathbf{P}_j (\nabla l(\boldsymbol{\theta}))_j\|_{\infty} \leq c,$$

for some finite c . Choose a $\lambda_0 > c$, and let $\hat{\theta}_1, \dots, \hat{\theta}_p$ be the solution to Equation (18) with $\alpha = 1$ and $\lambda = \lambda_0$. We claim that $\mathbf{D}_j \mathbf{P}_j \hat{\theta}_j = 0$, and $\tilde{\theta}_j = \hat{\theta}_j$ for $j = 1, \dots, p$. The optimality condition for Equation (18), evaluated at $\tilde{\theta}$, is

$$(\nabla l(\tilde{\theta}))_j + \lambda_0 \mathbf{P}_j^T \mathbf{D}_j^T \gamma(\mathbf{D}_j \mathbf{P}_j \hat{\theta}_j) = 0, \quad (\text{C.2})$$

where

$$\gamma(\mathbf{D}_j \mathbf{P}_j \hat{\theta}_j)_i = \begin{cases} +1, & (\mathbf{D}_j \mathbf{P}_j \hat{\theta}_j)_i > 0 \\ -1, & (\mathbf{D}_j \mathbf{P}_j \hat{\theta}_j)_i < 0, \\ \{a: |a| \leq 1\}, & (\mathbf{D}_j \mathbf{P}_j \hat{\theta}_j)_i = 0 \end{cases} \quad (\text{C.3})$$

for $j = 1, \dots, p$. Multiplying $\mathbf{D}_j \mathbf{P}_j$ on both sides and recognizing that $\mathbf{D}_j \mathbf{D}_j^T$ is invertible, Equation (C.2) becomes

$$(\mathbf{D}_j \mathbf{D}_j^T)^{-1} \mathbf{D}_j \mathbf{P}_j (\nabla l(\tilde{\theta}))_j + \lambda_0 \gamma(\mathbf{D}_j \mathbf{P}_j \hat{\theta}_j) = 0.$$

Recall that $\lambda_0 > \max_j \sup_{\theta} \|(\mathbf{D}_j \mathbf{D}_j^T)^{-1} \mathbf{D}_j \mathbf{P}_j (\nabla l(\theta))_j\|_{\infty}$, so $\|\gamma(\mathbf{D}_j \mathbf{P}_j \hat{\theta}_j)\|_{\infty} < 1$. Then Equation (C.3) implies that $\mathbf{D}_j \mathbf{P}_j \hat{\theta}_j = 0$. Furthermore, notice that $\tilde{\theta}_j = \hat{\theta}_j$ for $j = 1, \dots, p$. This completes the proof of the claim.

Next, we want to show sufficiency. For any $\lambda \geq \max_j \|(\mathbf{D}_j \mathbf{D}_j^T)^{-1} \mathbf{D}_j \mathbf{P}_j (\nabla l(\tilde{\theta}))_j\|_{\infty}$, where $\tilde{\theta}$ is

the solution to Equation (C.1), take $\gamma_j = -\frac{(\mathbf{D}_j \mathbf{D}_j^T)^{-1} \mathbf{D}_j \mathbf{P}_j (\nabla l(\tilde{\theta}))_j}{\lambda}$. Clearly $\|\gamma\|_{\infty} \leq 1$. From Equation (C.2), we know that $(\nabla l(\tilde{\theta}))_j = -\lambda_0 \mathbf{P}_j^T \mathbf{D}_j^T \gamma(\mathbf{D}_j \mathbf{P}_j \tilde{\theta}_j)$. We can see that

$$\begin{aligned} (\nabla l(\tilde{\theta}))_j + \lambda \mathbf{P}_j^T \mathbf{D}_j^T \gamma_j &= (\nabla l(\tilde{\theta}))_j - \mathbf{P}_j^T \mathbf{D}_j^T (\mathbf{D}_j \mathbf{D}_j^T)^{-1} \mathbf{D}_j \mathbf{P}_j (\nabla l(\tilde{\theta}))_j \\ &= -\lambda_0 \mathbf{P}_j^T \mathbf{D}_j^T \gamma(\mathbf{D}_j \mathbf{P}_j \tilde{\theta}_j) + \lambda_0 \mathbf{P}_j^T \mathbf{D}_j^T (\mathbf{D}_j \mathbf{D}_j^T)^{-1} \mathbf{D}_j \mathbf{P}_j \mathbf{P}_j^T \mathbf{D}_j^T \gamma(\mathbf{D}_j \mathbf{P}_j \tilde{\theta}_j) \\ &= -\lambda_0 \mathbf{P}_j^T \mathbf{D}_j^T \gamma(\mathbf{D}_j \mathbf{P}_j \tilde{\theta}_j) + \lambda_0 \mathbf{P}_j^T \mathbf{D}_j^T \gamma(\mathbf{D}_j \mathbf{P}_j \tilde{\theta}_j) \\ &= 0. \end{aligned}$$

Thus, $\tilde{\theta}$ is the solution to Equation (18) corresponding to this λ .

We also want to show necessity. If $\tilde{\theta}$ solves Equation (18), then there exist $\gamma_{ij} \in [-1, 1]$, $i = 1, \dots, n$, such that $(\nabla l(\tilde{\theta}))_j + \lambda \mathbf{P}_j^T \mathbf{D}_j^T \gamma_j = 0$. This means that $(\mathbf{D}_j \mathbf{D}_j^T)^{-1} \mathbf{D}_j \mathbf{P}_j (\nabla l(\tilde{\theta}))_j + \lambda \gamma_j = 0$. Therefore, $\lambda \geq \max_j \|(\mathbf{D}_j \mathbf{D}_j^T)^{-1} \mathbf{D}_j \mathbf{P}_j (\nabla l(\tilde{\theta}))_j\|_{\infty}$. $\square$

Proposition 4. Let $\hat{\theta}_j$ denote the solution to Equation (18) with $\alpha = 0$. Then $\hat{\theta}_j = \mathbf{0}$ for $j = 1, \dots, p$ if and only if $\lambda \geq \lambda_{\max,2}$, where $\lambda_{\max,2} \equiv \max_j \{\|(\nabla l(\mathbf{0}))_j\|_2\}$.

Proof. The optimality condition is $(\nabla l(\hat{\theta}))_j + \lambda \gamma(\hat{\theta}_j) = \mathbf{0}$ for $j = 1, \dots, p$, where

$$\gamma(\hat{\theta}_j) = \begin{cases} \frac{\hat{\theta}_j}{\|\hat{\theta}_j\|_2} & \hat{\theta}_j \neq \mathbf{0} \\ \{\mathbf{a}: \|\mathbf{a}\|_2 \leq 1\} & \hat{\theta}_j = \mathbf{0} \end{cases}.$$

If $\hat{\theta}_j = \mathbf{0}$, then $(\nabla l(\mathbf{0}))_j + \lambda \gamma(\mathbf{0}_j) = \mathbf{0}$. Since $\|\gamma(\mathbf{0}_j)\|_2 = 1$, we have $\lambda \leq \|(\nabla l(\mathbf{0}))_j\|_2$ for $j = 1, \dots, p$. Thus, $\lambda \leq \lambda_{\max,2}$.

On the other hand, if $\lambda > \lambda_{\max,2}$, let $\gamma_j = -\frac{(\nabla l(\mathbf{0}))_j}{\lambda}$. Clearly $\|\gamma_j\|_2 = 1$, and also $(\nabla l(\mathbf{0}))_j + \lambda \gamma_j = \mathbf{0}$. Thus, $\hat{\theta}_j = \mathbf{0}$. $\square$

Proposition 5. Let $\hat{\theta}_j$ denote the solution to Equation (18) with $\alpha \in (0, 1)$. Then a sufficient condition for $\hat{\theta}_j = \mathbf{0}$, for $j = 1, \dots, p$, is $\lambda \leq \lambda_{\max,3}$, where $\lambda_{\max,3} \equiv \max_j \frac{\|(\nabla l(\mathbf{0}))_j\|_2}{1-\alpha}$.

Proof. The objective in Equation (18) is bounded below by

$$\min_{\theta} \left\{ l(\theta) + (1-\alpha)\lambda \sum_{j=1}^p \|\theta_j\|_2 \right\}. \quad (\text{C.4})$$

Repeating logic in Proposition 4 reveals that $\hat{\theta}_j = \mathbf{0}$ for $j = 1, \dots, p$ solves (C.4) if and only if $(1-\alpha)\lambda \leq \max_j \{\|(\nabla l(\mathbf{0}))_j\|_2\}$. Furthermore, we see from inspection of (18) and (C.4) that if $\hat{\theta}_j = \mathbf{0}$ solves (C.4), then $\hat{\theta}_j = \mathbf{0}$ also solves (18). Thus, a sufficient condition for the solution to (18) to equal zero is $\lambda \leq \lambda_{\max,3}$. $\square$

Appendix

Appendix D: Proximal Gradient Descent Algorithm to Solve Equation (23)

We now derive an algorithm to solve the optimization problem Equation (23), which allows for both discrete- and continuous-valued covariates in the Cox trend filtering problem.

Suppose that $\mathbf{X}_j$ is a discrete-valued covariate, $j \in \mathcal{A}^c$, with M_j levels. Let $X_{(m)j}$ be the m th unique value of $\mathbf{X}_j$, and let $\beta_{(m)j}$ be the corresponding coefficient value for $m = 1, \dots, M_j$. Note the constraint in Equation (23): $\theta_{ij} = \theta_{uj}$ if $X_{ij} = X_{uj}$. We can reparameterize θ_{ij} in terms of $\beta_{(m)j}$ with the relationship $\theta_{ij} = \beta_{(m)j}$ if $X_{ij} = X_{(m)j}$. Since $\beta_{(m)j}$ is not penalized in Equation (23), we can use gradient descent to solve for $\beta_{(m)j}$. By the chain rule, the gradient is of the form

$$\frac{\partial l(\theta)}{\partial \beta_{(m)j}} = \sum_{i: X_{ij} = X_{(m)j}} \frac{\partial l(\theta)}{\partial \theta_{ij}}.$$

Then we have the following gradient descent update for discrete covariate j :

$$\beta_{(m)j}^{(h+1)} \leftarrow \beta_{(m)j}^{(h)} - s \frac{\partial l(\theta)}{\partial \beta_{(m)j}},$$

$$\theta_{ij}^{(h+1)} \leftarrow \beta_{(m)j}^{(h+1)} \quad \text{for } i \text{ such that } X_{ij} = X_{(m)j},$$

which can be written in terms of θ_{ij} :

$$\theta_{ij}^{(h+1)} \leftarrow \theta_{ij}^{(h)} - s \sum_{i: X_{ij} = X_{(m)j}} \left(\nabla l(\theta^{(h)}) \right)_{ij} \quad \text{for } i \text{ such that } X_{ij} = X_{(m)j},$$

for $m = 1, \dots, M_j$. For continuous covariates $\mathbf{X}_j$ where $j \in \mathcal{A}$, we can use proximal gradient descent as in Algorithm 1 to solve for θ_j . Details of solving Equation (23) are provided in Algorithm 2.

Appendix

Appendix E: Additional Simulation Results

In this section, we investigate additional simulation scenarios beyond those considered in Section 4.

E.1. Simulation With 4% and 50% Censoring

To investigate whether the results in Section 4 are affected by the proportion of observations censored, we consider uninformative censoring with an Exponential(0.01) and Exponential(1) distribution, such that

Algorithm 2

Proximal Gradient for Cox Trend Filtering Model with Discrete and Continuous Covariates
(23)

-
1. Initialize $\theta_j^{(0)} \leftarrow \mathbf{0}$, and set $\theta_j^{\text{new}} \leftarrow \theta_j^{(0)}$, for $j = 1, \dots, p$.
 2. Select a scalar step size $s = 25$, and a constant $\beta = 0.8$.
 3. For $h = 0, 1, 2, \dots$ until convergence:

$$(a) \text{ While } l(\theta^{\text{new}}) \geq l(\theta^{(h)}) + (\nabla l(\theta^{(h)}))^T (\theta^{\text{new}} - \theta^{(h)}) + \frac{1}{2s} \|\theta^{\text{new}} - \theta^{(h)}\|_2^2:$$

- i. Set $s \leftarrow \beta s$.
- ii. For each continuous covariate index $j \in \mathcal{A}$:

A. Take a gradient step,

$$\tilde{\theta}_j \leftarrow \theta_j^{(h)} - s \left(\nabla l(\theta^{(h)}) \right)_j.$$

B. Use an existing solver, such as the `glmgen` package in R (Arnold, Sadhanala, and Tibshirani 2014), to compute

$$\theta_j^{\text{new}} \leftarrow \underset{\theta_j}{\operatorname{argmin}} \left\{ \frac{1}{2} \|\tilde{\theta}_j - \theta_j\|_2^2 + s\lambda \alpha \left\| \mathbf{D}^{(\mathbf{P}_j \mathbf{X}_j, k+1)} \mathbf{P}_j \theta_j \right\|_1 \right\}.$$

C. Soft-scale the estimate,

$$\theta_j^{\text{new}} \leftarrow \left(1 - \frac{(1 - \alpha)s\lambda}{\|\theta_j^{\text{new}}\|_2} \right)_+ \theta_j^{\text{new}}.$$

(b) For each discrete covariate index $j \in \mathcal{A}^c$:

- i. For $m = 1, 2, \dots, M_j$, where M_j is the number of unique values of $\mathbf{X}_j$:

$$\theta_{uj}^{\text{new}} \leftarrow \theta_{uj}^{(h)} - s \sum_{u: X_{uj} = X_{(m)j}} \left(\nabla l(\theta^{(h)}) \right)_{uj}$$

for u such that $X_{uj} = X_{(m)j}$.

(c) Set $\theta^{(h+1)} \leftarrow \theta^{\text{new}}$.

approximately 4% and 50% of the subjects are censored, respectively. Table E.1 and E.2 show the results for 4% and 50% censoring, respectively, in the low-dimensional setting

with $n = 100$ and $p = 4$ (see Section 4 for additional simulation details). Compared with Table 2, increasing censoring leads to an increase in parameter fit errors. As before, constant trend filtering outperforms GAM and SGL in scenarios 1 and 4, while linear trend filtering and GAM perform the best in scenarios 2 and 3.

E.2. Simulation With Discrete- and Continuous-Valued Covariates

To investigate the performance of Algorithm 2 with discrete covariates, we generate survival data from Equation (24). We generated four continuous-valued covariates as in Section 4, and we generated ten discrete-valued covariates that take on values of 0 and 1 with probability 0.5. For a discrete-valued covariate x_j , we define $f_j(x_j = 0) = -0.5$ and $f_j(x = 1) = -0.5$. There are no noise covariates. We consider a sample size of $n = 100$, and uninformative censoring with 4.5% of the subjects censored. Additional simulation details can be found in Section 4. Table E.3 shows the performance of constant trend filtering, linear trend filtering, GAM, and SGL. Similar to Table E.2, constant trend filtering performs better in scenarios 1 and 4, and linear trend filtering and GAM perform better in scenarios 2 and 3.

Table E.1.

Results of the simulation study described in Section E.1 with 4% of subjects censored.

	Loss	Parameter fit error	Number of knots
<i>Scenario 1</i>			
Constant trend filtering, $\alpha = 1$	3.08(0.01)	77.4(2.4)	24.8(0.6)
Linear trend filtering, $\alpha = 1$	3.15(0.01)	94.7(2.7)	15.2(0.7)
GAM	3.14(0.01)	94.2(1.8)	
SGL, $\alpha = 1$	3.13(0.01)	116.8(2.7)	19.8(0.4)
<i>Scenario 2</i>			
Constant trend filtering, $\alpha = 1$	3.11(0.01)	80.4(3.0)	34.3(0.7)
Linear trend filtering, $\alpha = 1$	3.00(0.01)	39.5(1.4)	15.3(0.4)
GAM	3.02(0.01)	40.7(2.1)	
SGL, $\alpha = 1$	3.02(0.01)	59.6(2.1)	18.4(0.4)
<i>Scenario 3</i>			
Constant trend filtering, $\alpha = 1$	3.18(0.01)	106.7(3.6)	41.0(0.7)
Linear trend filtering, $\alpha = 1$	2.99(0.01)	36.8(1.3)	15.7(0.3)
GAM	3.02(0.01)	41.43(1.6)	
SGL, $\alpha = 1$	3.01(0.01)	66.3(1.9)	16.6(0.4)
<i>Scenario 4</i>			
Constant trend filtering, $\alpha = 1$	3.14(0.01)	106.0(2.8)	21.9(0.7)
Linear trend filtering, $\alpha = 1$	3.20(0.01)	110.0(3.0)	12.6(0.8)
GAM	3.20(0.01)	119.7(2.3)	
SGL, $\alpha = 1$	3.17(0.01)	131.0(2.5)	20.8(0.3)

NOTE: Details are as in Table 2.

Table E.2.

Results of the simulation study described in Section E.1 with 50% of subjects censored.

	Loss	Parameter fit error	Number of knots
<i>Scenario 1</i>			
Constant trend filtering, $\alpha = 1$	1.72(0.01)	101.3(2.8)	22.4(0.6)
Linear trend filtering, $\alpha = 1$	1.66(0.01)	117.8(3.0)	16.6(1.2)
GAM	1.74(0.02)	115.1(2.6)	
SGL, $\alpha = 1$	1.76(0.01)	167.9(3.6)	11.0(0.3)
<i>Scenario 2</i>			
Constant trend filtering, $\alpha = 1$	1.73(0.01)	117.8(3.4)	27.8(0.6)
Linear trend filtering, $\alpha = 1$	1.66(0.02)	72.0(2.9)	19.6(1.2)
GAM	1.66(0.02)	56.9(2.8)	
SGL, $\alpha = 1$	1.68(0.01)	117.8(4.1)	10.6(0.3)
<i>Scenario 3</i>			
Constant trend filtering, $\alpha = 1$	1.77(0.02)	144.6(4.0)	32.6(0.7)
Linear trend filtering, $\alpha = 1$	1.66(0.01)	68.3(2.4)	19.5(1.0)
GAM	1.66(0.02)	65.2(2.4)	
SGL, $\alpha = 1$	1.68(0.01)	112.9(3.3)	10.8(0.3)
<i>Scenario 4</i>			
Constant trend filtering, $\alpha = 1$	1.67(0.01)	117.4(3.1)	17.1(0.5)
Linear trend filtering, $\alpha = 1$	1.71(0.02)	132.4(2.7)	11.3(1.3)
GAM	1.70(0.02)	132.7(2.6)	
SGL, $\alpha = 1$	1.74(0.02)	186.2(3.8)	10.4(0.3)

NOTE: Details are as in Table 2.

Table E.3.

Results of the simulation study described in Section E.2 with discrete- and continuous-valued covariates.

	Loss	Parameter fit error	Number of knots
<i>Scenario 1</i>			
Constant trend filtering, $\alpha = 1$	2.94(0.01)	115.2(3.6)	26.7(0.5)
Linear trend filtering, $\alpha = 1$	3.06(0.01)	140.9(3.6)	21.1(1.3)
GAM	3.05(0.01)	124.6(2.4)	
SGL, $\alpha = 1$	3.05(0.01)	197.4(4.7)	12.9(0.2)
<i>Scenario 2</i>			
Constant trend filtering, $\alpha = 1$	3.05(0.02)	132.1(4.6)	34.6(0.6)
Linear trend filtering, $\alpha = 1$	2.93(0.01)	77.2(2.6)	18.2(0.8)
GAM	2.91(0.02)	74.0(4.9)	
SGL, $\alpha = 1$	2.97(0.01)	142.0(5.1)	12.7(0.2)
<i>Scenario 3</i>			
Constant trend filtering, $\alpha = 1$	3.14(0.01)	176.2(4.5)	43.1(0.8)
Linear trend filtering, $\alpha = 1$	2.88(0.01)	77.3(2.8)	16.9(0.4)

	Loss	Parameter fit error	Number of knots
GAM	2.96(0.01)	84.5(3.8)	
SGL, $\alpha = 1$	2.91(0.01)	141.7(4.8)	12.2(0.2)
<i>Scenario 4</i>			
Constant trend filtering, $\alpha = 1$	3.03(0.01)	149.7(3.2)	21.8(0.7)
Linear trend filtering, $\alpha = 1$	3.09(0.01)	154.4(3.5)	16.0(1.5)
GAM	3.11(0.02)	146.8(2.8)	
SGL, $\alpha = 1$	3.09(0.01)	239.9(5.5)	13.0(0.2)

NOTE: Details are as in Table 2.

References

- Arnold T, Sadhanala V, and Tibshirani R (2014), glmgen: Fast Algorithms for Generalized Lasso Problems, R package version 0.0.3. [3,4,11]
- Beck A, and Teboulle M (2009), “A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems,” *SIAM Journal on Imaging Sciences*, 2, 183–202. [3,4]
- Boyd S, and Vandenberghe L (2004), *Convex Optimization*, Cambridge: Cambridge University Press [3]
- Breslow N (1974), “Covariance Analysis of Censored Survival Data,” *Biometrics*, 30, 89–99. [3] [PubMed: 4813387]
- Bühlmann P (2013), “Statistical Significance in High-Dimensional Linear Models,” *Bernoulli*, 19, 1212–1242. [9]
- Chaturvedi N, de Menezes RX, and Goeman JJ (2014), “Fused Lasso Algorithm for Cox’s Proportional Hazards and Binomial Logit Models With Application to Copy Number Profiles,” *Biometrical Journal*, 56, 477–492. [2] [PubMed: 24496763]
- Cox DR (1972), “Regression Models and Life Tables (with discussion),” *Journal of the Royal Statistical Society*, 34, 187–220. [1]
- Daubechies I (1992), *Ten Lectures on Wavelets*, Philadelphia, PA: SIAM [1]
- Efron B (1986), “How Biased Is the Apparent Error Rate of a Prediction Rule?,” *Journal of the American Statistical Association*, 81, 461–470. [8]
- Fleming TR, and Harrington DP (1991), *Counting Processes and Survival Analysis*, Hoboken, NJ: Wiley [6]
- Friedman JH (1991), “Multivariate Adaptive Regression Splines,” *The Annals of Statistics*, 19, 1–67. [1]
- Gordon D, Taddei-Peters W, Mascette A, Antman M, Kaufmann PG and Lauer MS (2013), “Publication of Trials Funded by the National Heart, Lung, and Blood Institute,” *New England Journal of Medicine*, 369, 1926–1934. [7] [PubMed: 24224625]
- Gray RJ (1992), “Flexible Methods for Analyzing Survival Data Using Splines, With Applications to Breast Cancer Prognosis,” *Journal of the American Statistical Association*, 87, 942–951. [2]
- Hastie TJ, and Tibshirani RJ (1990a), *Generalized Additive Models* (Vol. 43), Boca Raton, FL: CRC Press [1,2,8]
- (1990b), “Exploring the Nature of Covariate Effects in the Proportional Hazards Model,” *Biometrics*, 46, 1005–1016. [2] [PubMed: 1964808]
- Heagerty PJ, Lumley T, and Pepe MS (2000), “Time-Dependent ROC Curves for Censored Survival Data and a Diagnostic Marker,” *Biometrics*, 56, 337–344. [7] [PubMed: 10877287]
- Huang J, Sun T, Ying Z, Yu Y, and Zhang C-H (2013), “Oracle Inequalities for the Lasso in the Cox Model,” *The Annals of Statistics*, 41, 1142 [8] [PubMed: 24086091]
- Huang J, Wei F, and Ma S (2012), “Semiparametric Regression Pursuit,” *Statistica Sinica*, 22, 1403 [1] [PubMed: 23559831]

- Hyun S, G'Sell M, and Tibshirani RJ (2016), "Exact Post-Selection Inference for Changepoint Detection and Other Generalized Lasso Problems," arXiv preprint arXiv:1606.03552. [9]
- Javanmard A, and Montanari A (2014a), "Confidence Intervals and Hypothesis Testing for High-Dimensional Regression," *The Journal of Machine Learning Research*, 15, 2869–2909. [9]
- (2014b), "Hypothesis Testing in High-Dimensional Regression Under the Gaussian Random Design Model: Asymptotic Theory," *IEEE Transactions on Information Theory*, 60, 6522–6554. [9]
- Kim S-J, Koh K, Boyd S, and Gorinevsky D (2009), " ℓ_1 Trend Filtering," *SIAM Review*, 51, 339–360. [1,2]
- Kong S, and Nan B (2014), "Non-Asymptotic Oracle Inequalities for the High-Dimensional Cox Regression Via Lasso," *Statistica Sinica*, 24, 25 [8] [PubMed: 24516328]
- Kooperberg C, Stone CJ, and Truong YK (1995), "Hazard Regression," *Journal of the American Statistical Association*, 90, 78–94. [2]
- Lee JD, Sun DL, Sun Y, and Taylor JE (2016), "Exact Post-Selection Inference, With Application to the Lasso," *The Annals of Statistics*, 44, 907–927. [9]
- Leng C, Lin Y, and Wahba G (2006), "A Note on the Lasso and Related Procedures in Model Selection," *Statistica Sinica*, 16, 1273–1284. [6]
- Lin Y, and Zhang HH (2006), "Component Selection and Smoothing in Multivariate Nonparametric Regression," *The Annals of Statistics*, 34, 2272–2297. [2]
- Lockhart R, Taylor J, Tibshirani RJ, and Tibshirani R (2014), "A Significance Test for the Lasso," *The Annals of Statistics*, 42, 413 [9] [PubMed: 25574062]
- Lou Y, Bien J, Caruana R, and Gehrke J (2016), "Sparse Partially linear Additive Models," *Journal of Computational and Graphical Statistics*, 25, 1126–1140. [1,2]
- Meier L, Van de Geer S, and Bühlmann P (2009), "High-Dimensional Additive Modelling," *The Annals of Statistics*, 37, 3779–3821. [2]
- Meinshausen N, and Bühlmann P (2006), "High-Dimensional Graphs and Variable Selection With the Lasso," *The Annals of Statistics*, 34, 1436–1462. [6]
- Meir A, and Drton M (2017), "Tractable Post-Selection Maximum Likelihood Inference for the Lasso," arXiv preprint arXiv:1705.09417. [9]
- Molinari N, Daures J-P, and Durand J-F (2001), "Regression Splines for Threshold Selection in Survival Data Analysis," *Statistics in Medicine*, 20, 237–247. [2] [PubMed: 11169599]
- Nadaraya EA (1964), "On Estimating Regression," *Theory of Probability & Its Applications*, 9, 141–142. [1]
- O'Sullivan F (1988), "Nonparametric Estimation of Relative Risk Using Splines and Cross-Validation," *SIAM Journal on Scientific and Statistical Computing*, 9, 531–542. [2]
- Parikh N, and Boyd S (2014), "Proximal Algorithms," *Foundations and Trends® in Optimization*, 1, 127–239. [3,9,10]
- Park T, and Casella G (2008), "The Bayesian Lasso," *Journal of the American Statistical Association*, 103, 681–686. [9]
- Petersen A, and Witten D (2019), "Data-Adaptive Additive Modeling," *Statistics in Medicine*, 38, 583–600. [1] [PubMed: 30010200]
- Petersen A, Witten D, and Simon N (2016), "Fused Lasso Additive Model," *Journal of Computational and Graphical Statistics*, 25, 1005–1025. [1,2,3,4,5,8] [PubMed: 28239246]
- Ramdas A, and Tibshirani RJ (2016), "Fast and Flexible ADMM Algorithms for Trend Filtering," *Journal of Computational and Graphical Statistics*, 25, 839–858. [4,6]
- Ravikumar P, Lafferty J, Liu H, and Wasserman L (2009), "Sparse Additive Models," *Journal of the Royal Statistical Society, Series B*, 71, 1009–1030. [1,2]
- Sadhanala V, and Tibshirani RJ (2017), "Additive Models With Trend Filtering," arXiv preprint arXiv:1702.05037. [1,2,8]
- Simon N, Friedman J, Hastie T, and Tibshirani R (2013), "A Sparse-Group Lasso," *Journal of Computational and Graphical Statistics*, 22, 231–245. [5]
- Simon N, and Tibshirani R (2012), "Standardization and the Group Lasso Penalty," *Statistica Sinica*, 22, 983 [3] [PubMed: 26257503]

- Sleeper LA, and Harrington DP (1990), “Regression Splines in the Cox Model With Application to Covariate Effects in Liver Disease,” *Journal of the American Statistical Association*, 85, 941–949. [2]
- Tibshirani RJ (2014), “Adaptive Piecewise Polynomial Estimation via Trend Filtering,” *The Annals of Statistics*, 42, 285–323. [1,2,8]
- Tibshirani RJ, 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*, 67, 91–108. [1]
- Tibshirani RJ, Taylor J, Lockhart R, and Tibshirani R (2016), “Exact Post-Selection Inference for Sequential Regression Procedures,” *Journal of the American Statistical Association*, 111, 600–620. [9]
- Van Buuren S (2018), *Flexible Imputation of Missing Data*, Boca Raton, FL: Chapman and Hall/CRC [6]
- Van de Geer S, Bühlmann P, Ritov Y, Dezeure R (2014), “On Asymptotically Optimal Confidence Regions and Tests for High-Dimensional Models,” *The Annals of Statistics*, 42, 1166–1202. [9]
- Wahba G (1990), *Spline Models for Observational Data*, Philadelphia, PA: SIAM [1]
- Watson GS (1964), “Smooth Regression Analysis,” *Sankhyā : The Indian Journal of Statistics, Series A*, 26, 359–372. [1]
- Wood S (2006), *Generalized Additive Models: An Introduction With R*, Boca Raton, FL: CRC Press [2]
- Wood S, and Wood MS (2016), “Package ‘MGCV’,” R package version, pp. 1–7. [5]
- Wu Y, and Stefanski LA (2015), “Automatic Structure Recovery for Additive Models,” *Biometrika* 102, 381–395. [1] [PubMed: 26146407]
- Yuan M, and Lin Y (2006), “Model Selection and Estimation in Regression With Grouped Variables,” *Journal of the Royal Statistical Society, Series B*, 68, 49–67. [3]
- Zhang C-H, and Zhang SS (2014), “Confidence Intervals for Low Dimensional Parameters in High Dimensional Linear Models,” *Journal of the Royal Statistical Society, Series B*, 76, 217–242. [9]
- Zhang HH, Cheng G, and Liu Y (2011), “Linear or Nonlinear? Automatic Structure Discovery for Partially Linear Models,” *Journal of the American Statistical Association*, 106, 1099–1112. [1] [PubMed: 22121305]

Figure 1.

In the simulation study in Section 4, the functions $f_1, \dots, f_4$ are piece-wise constant (Scenario 1), smooth (Scenario 2), piece-wise linear (Scenario 3), and highly variable in some areas and constant in other areas (Scenario 4).

Figure 2.

Comparison between constant trend filtering (left panel), linear trend filtering (center panel), and GAM (right panel) fits on scenario 4 of the simulation study, described in Section 4. The thick black line indicates the true underlying function. 15 fits, each from a separate simulated dataset, are shown in each plot. Constant trend filtering can better adapt to the local smoothness of the underlying functions relative to linear trend filtering and GAM.

Figure 3.

Estimated log hazard ratios of constant trend filtering (dashed), linear trend filtering (dotted), GAM (dotdash), and Cox model with linear predictors (solid) for the PBC dataset. Eight continuous variables are shown: age (years), serum bilirubin (mg/dl), serum cholesterol (mg/dl), serum albumin (g/dl), urine copper (ug/day), alkaline phosphatase (U/liter), aspartate transaminase (U/ml), and triglycerides (mg/dl). For constant trend filtering, changes in the log hazard ratio occur only at discrete values of the features. For the other methods, the log hazard ratio is a continuous function of the features.

Figure 4.

Estimated log hazard ratios of constant trend filtering (dashed), linear trend filtering (dotted), GAM (dotdash), and Cox model with linear predictors (solid) for the publication dataset. Three continuous variables are shown: sample size, budget (in millions of dollars), and impact factor. For constant trend filtering, changes in the log hazard ratio occur only at discrete values of the features. For the other methods, the log hazard ratio is a continuous function of the features.

Table 1.

Mean timing results, in seconds, for a single simulated dataset in the simulation study described in Section 4.

	Low-dimensional	High-dimensional
<i>Scenario 1</i>		
Constant trend filtering	0.50(0.01)	15.0(0.6)
Linear trend filtering	2.35(0.02)	196.3(5.3)
GAM	0.51(0.01)	
SGL	0.44(0.01)	37.5(1.4)
<i>Scenario 2</i>		
Constant trend filtering	0.50(0.01)	15.4(0.6)
Linear trend filtering	2.51(0.03)	198.5(5.2)
GAM	0.54(0.01)	
SGL	0.48(0.01)	34.8(1.3)
<i>Scenario 3</i>		
Constant trend filtering	0.51(0.01)	14.5(0.6)
Linear trend filtering	2.54(0.03)	197.8(5.2)
GAM	0.48(0.01)	
SGL	0.46(0.01)	37.3(1.3)
<i>Scenario 4</i>		
Constant trend filtering	0.49(0.01)	15.1(0.6)
Linear trend filtering	2.64(0.03)	199.6(5.4)
GAM	0.53(0.01)	
SGL	0.43(0.01)	38.4(1.2)

NOTE: The SE of the mean is displayed in parentheses. Results for both the low- and high-dimensional settings are displayed. GAM was run only in the low-dimensional setting.

Table 2.

Results of the simulation study described in Section 4, in the low-dimensional setting with 25% of subjects censored.

	Loss	Parameter fit error	Number of knots
<i>Scenario 1</i>			
Constant trend filtering, $\alpha = 1$	2.51(0.01)	80.6(2.3)	25.4(0.6)
Linear trend filtering, $\alpha = 1$	2.44(0.01)	104.3(2.5)	14.6(0.7)
GAM	2.56(0.01)	104.0(1.9)	
SGL, $\alpha = 1$	2.55(0.01)	146.2(2.9)	12.5(0.2)
<i>Scenario 2</i>			
Constant trend filtering, $\alpha = 1$	2.55(0.01)	95.1(2.9)	30.9(0.6)
Linear trend filtering, $\alpha = 1$	2.44(0.01)	48.6(1.4)	13.8(0.3)
GAM	2.43(0.01)	43.7(1.7)	
SGL, $\alpha = 1$	2.47(0.01)	88.8(3.1)	12.2(0.3)
<i>Scenario 3</i>			
Constant trend filtering, $\alpha = 1$	2.57(0.01)	116.84(3.8)	38.1(0.8)
Linear trend filtering, $\alpha = 1$	2.42(0.01)	48.3(1.9)	15.3(0.3)
GAM	2.43(0.01)	49.5(1.5)	
SGL, $\alpha = 1$	2.46(0.01)	90.4(2.6)	11.9(0.2)
<i>Scenario 4</i>			
Constant trend filtering, $\alpha = 1$	2.51(0.01)	106.4(2.8)	19.3(0.6)
Linear trend filtering, $\alpha = 1$	2.57(0.01)	116.7(3.1)	11.0(0.8)
GAM	2.50(0.01)	126.7(2.4)	
SGL, $\alpha = 1$	2.59(0.01)	161.1(3.6)	12.8(0.3)

NOTE: The means (across 100 simulated datasets) of the validation set loss Equation (16), the parameter fit error $\sum_{j=1}^p \|\theta_j - \hat{\theta}_j\|_2^2$, and the number of knots are displayed. SEs are shown in parentheses.

Table 3.

Results of the simulation study described in Section 4, in the high-dimensional setting with 25% of subjects censored.

	Loss	Parameter fit error	Number of knots	Proportion of nonzero features
<i>Scenario 1</i>				
Constant trend filtering	2.59(0.01)	174.2(3.9)	98.4(3.6)	0.20(0.01)
Linear trend filtering	2.63(0.01)	202.1(3.6)	81.4(2.9)	0.21(0.01)
SGL	2.64(0.01)	198.6(3.5)	80.4(2.3)	0.20(0.01)
<i>Scenario 2</i>				
Constant trend filtering	2.63(0.01)	178.8(4.1)	138.6(5.0)	0.21(0.01)
Linear trend filtering	2.54(0.01)	132.6(3.0)	74.2(2.5)	0.22(0.01)
SGL	2.58(0.02)	160.4(4.1)	79.6(2.7)	0.22(0.01)
<i>Scenario 3</i>				
Constant trend filtering	2.64(0.01)	209.5(4.9)	156.0(5.7)	0.18(0.01)
Linear trend filtering	2.55(0.01)	148.7(3.3)	72.5(2.1)	0.22(0.01)
SGL	2.58(0.01)	174.3(4.5)	77.6(2.9)	0.23(0.01)
<i>Scenario 4</i>				
Constant trend filtering	2.60(0.01)	177.3(3.4)	78.7(3.5)	0.18(0.01)
Linear trend filtering	2.64(0.01)	206.2(3.7)	83.6(3.0)	0.21(0.01)
SGL	2.65(0.01)	223.7(3.7)	78.3(2.4)	0.19(0.01)

NOTE: Details are as in Table 2. Here 4% of the features are truly nonzero.

Table 4.

Performance of each method on the validation set of the PBC data, averaged over 100 data splits.

	Loss	AUC	Time
Constant trend filtering	1.41(0.16)	0.840(0.035)	1.70
Linear trend filtering	1.46(0.19)	0.844(0.042)	4.28
GAM	1.56(0.30)	0.849(0.036)	1.80
Cox w/ linear predictors	1.51(0.21)	0.842(0.040)	0.007

NOTE: We report the mean (and SD) of the loss, as well as the AUC at the median survival time. We also report the computation time of each method in seconds.

Table 5.

Performance of each method on the validation set of the publication data, averaged over 100 data splits.

	Loss	AUC	Time
Constant trend filtering	1.98(0.16)	0.882(0.038)	0.47
Linear trend filtering	2.02(0.15)	0.875(0.035)	1.05
GAM	2.28(0.74)	0.868(0.047)	1.07
Cox w/ linear predictors	2.25(0.48)	0.857(0.049)	0.004

NOTE: We report the mean (and SD) of the loss, as well as the AUC at the median survival time. We also report the computation time of each method in seconds.