TOP (2024) 32:57–82
https://doi.org/10.1007/s11750-023-00658-4

ORIGINAL PAPER

Cutting-plane algorithm for estimation of sparse Cox proportional hazards models

Hiroki Saishu[1] · Kota Kudo[1] · Yuichi Takano[2]

Received: 24 August 2022 / Accepted: 30 June 2023 / Published online: 21 July 2023

Abstract

Survival analysis is a family of statistical methods for analyzing event occurrence times. We adopt a mixed-integer optimization approach to estimation of sparse Cox proportional hazards (PH) models for survival analysis. Specifically, we propose a high-performance cutting-plane algorithm based on a reformulation of our sparse estimation problem into a bilevel optimization problem. This algorithm solves the upper-level problem using cutting planes that are generated from the dual lower-level problem to approximate an upper-level nonlinear objective function. To solve the dual lower-level problem efficiently, we devise a quadratic approximation of the Fenchel conjugate of the loss function. We also develop a computationally efficient least-squares method for adjusting quadratic approximations to fit each dataset. Computational results demonstrate that our method outperforms regularized estimation methods in terms of accuracy for both prediction and subset selection especially for low-dimensional datasets. Moreover, our quadratic approximation of the Fenchel conjugate function accelerates the cutting-plane algorithm and maintains high generalization performance of sparse Cox PH models.

Keywords Cox model · Sparse estimation · Cutting-plane algorithm · Mixed-integer optimization · Fenchel conjugate · Survival analysis

Mathematics Subject Classification 90C11 · 90C25 · 90C90 · 62F07 · 62N02

✉ Yuichi Takano
ytakano@sk.tsukuba.ac.jp

[1] Graduate School of Science and Technology, University of Tsukuba, 1-1-1 Tennodai, Tsukuba-shi, Ibaraki 305-8573, Japan

[2] Institute of Systems and Information Engineering, University of Tsukuba, 1-1-1 Tennodai, Tsukuba-shi, Ibaraki 305-8573, Japan

1 Introduction

1.1 Background

Survival analysis (Klein and Moeschberger 2003) is a family of statistical methods for analyzing survival time, defined as the length of time between the start of an observation and the occurrence of the event of interest. Successful applications of survival analysis can be found in various real-world domains (Wang et al. 2019), such as gene expression analyses (Van De Vijver et al. 2002; Van Wieringen et al. 2009), customer relationship management (Rosset et al. 2003; Van den Poel and Larivière 2004), and credit risk evaluations (Demyanyk and Hasan 2010; Lane et al. 1986). One of the main challenges inherent to time-to-event data is the presence of censored instances, which do not experience occurrence of the event before the end of the observation period. Tailored statistical methods are widely used for analyzing time-to-event data with censored instances (Klein and Moeschberger 2003; Lee and Lim 2019; Wang et al. 2019). These methods can be categorized into three types of statistical models (Wang et al. 2019): parametric, nonparametric, and semiparametric models.

A primary purpose of survival analysis is to estimate a survival (or hazard) function, which represents the probability that an event of interest has not occurred by a certain time. Parametric models assume that the survival time follows a specific probability distribution (e.g., exponential, Weibull, logistic, or normal), whose parameters are tuned using tobit regression (Tobin 1958), Buckley–James regression (Buckley and James 1979), or an accelerated failure time (AFT) model (Saikia and Barman 2017). By contrast, nonparametric models estimate a survival function without such distributional assumptions. These include the Kaplan–Meier estimator (Kaplan and Meier 1958), the Nelson–Aalen estimator (Aalen 1978; Nelson 1972), and life-table analysis (Cutler and Ederer 1958). Semiparametric models, which are a hybrid of parametric and nonparametric models, can provide estimates that are more flexible than those from parametric models, and more stable than those from nonparametric models. Recently, machine learning techniques have been applied to survival analysis as well (Katzman et al. 2018; Wang et al. 2019).

1.2 Related work

We focus on estimation of sparse Cox proportional hazards (PH) models (Cox 1972), which are the most commonly used semiparametric method for survival analysis. The Cox PH model can examine how multiple features (explanatory variables) affect survival times. There are several strategies for selecting relevant features for the Cox PH model (Bradburn et al. 2003; Clark et al. 2003). In particular, various regularization methods have been applied to estimation of sparse Cox PH models. These include lasso (L_1-regularization) (Gui and Li 2005; Li et al. 2022; Tibshirani 1997), SCAD (Fan and Li 2002), adaptive lasso (Zhang and Lu 2007), correlation-based regularization (Vinzamuri and Reddy 2013), and elastic net (Goeman 2010; Park and Hastie 2007; Simon et al. 2011). However, these regularized estimation methods, which produce biased estimates due to the regularization term, are likely to yield low-quality solutions.

We adopt a mixed-integer optimization (MIO) approach to address the estimation of sparse Cox PH models. First proposed for linear regression in the 1970s (Arthanari and Dodge 1981), this approach has recently gained increased attention due to advances in optimization algorithms and computer hardware (Bertsimas et al. 2016; Cozad et al. 2014; Hastie et al. 2020; Konno and Yamamoto 2009; Ustun and Rudin 2016). In contrast to many heuristic optimization algorithms, the MIO approach has the advantage of selecting the best subset of features with respect to given criterion functions (Miyashiro and Takano 2015a, b; Park and Klabjan 2020; Takano and Miyashiro 2020). MIO-based methods for sparse estimation have been extended to logistic regression (Bertsimas and King 2017; Sato et al. 2016), ordinal regression (Naganuma et al. 2019; Sato et al. 2017), count regression (Saishu et al. 2021), support vector machine (Maldonado et al. 2014; Tamura et al. 2022), dimensionality reduction (Berk and Bertsimas 2019; Watanabe et al. 2023), and elimination of multicollinearity (Bertsimas and King 2016; Bertsimas and Li 2020; Tamura et al. 2017, 2019).

Bertsimas et al. (2021) recently proposed a high-performance cutting-plane algorithm that exactly solves MIO problems for sparse binary classification. They reformulated the problem as a bilevel optimization problem comprising lower- and upper-level problems. To solve the upper-level problem, its nonlinear objective function is iteratively approximated by generating cutting planes from solutions to the lower-level problem, based on the strong duality theory. Kamiya et al. (2019) extended the cutting-plane algorithm to the multinomial logit model for multiclass classification. They also devised a quadratic approximation of the lower-level objective function, which improved the computational efficiency of the cutting-plane algorithm and the generalization performance of resultant classification models.

1.3 Contribution

Our goal in this paper is to develop a high-performance cutting-plane algorithm for estimation of sparse Cox PH models. To our knowledge, we are the first to apply a cutting-plane algorithm (Bertsimas et al. 2021; Kamiya et al. 2019) to sparse estimation for survival analysis. Using the Fenchel conjugate (Wilson et al. 2021) of the loss function, we first derive a dual formulation of the L_2-regularized estimation of the Cox PH model. We next formulate a bilevel optimization problem for estimation of sparse Cox PH models. To solve the upper-level problem, we design a cutting-plane algorithm in which the dual lower-level problem is repeatedly solved to generate cutting planes for approximating an upper-level nonlinear objective function. Moreover, we devise a quadratic approximation of the Fenchel conjugate function to accelerate the cutting-plane algorithm. We also implement a computationally efficient least-squares method to allow calibration of quadratic approximations to each dataset.

We assess the efficacy of our method through computational experiments using synthetic and real-world datasets. With the synthetic datasets, our method outperforms regularized estimation methods in terms of accuracy for both prediction and subset selection especially for low-dimensional datasets. Moreover, our quadratic approximation of the Fenchel conjugate function accelerates the cutting-plane algorithm and maintains high generalization performance of sparse Cox PH models. Application to real-world datasets demonstrates that our method is well-suited to survival analysis in various real-world domains.

1.4 Notation

Throughout this paper, we denote the set of consecutive integers as $[n] := \{1, 2, \dots, n\}$. We write a p-dimensional column vector as $\boldsymbol{x} := (x_j)_{j \in [p]} \in \mathbb{R}^p$, and an $n \times p$ matrix as $\boldsymbol{X} := (x_{ij})_{(i,j) \in [n] \times [p]} \in \mathbb{R}^{n \times p}$.

2 Problem formulation

This section presents optimization formulations for estimation of sparse Cox PH models.

2.1 Cox proportional hazards model

Suppose that we are given a dataset $\{(t_i, \delta_i, \boldsymbol{x}_i) \mid i \in [n]\}$ consisting of n instances. Here, $t_i \in \mathbb{R}_+$ is the observed (event or censoring) time to be predicted, $\delta_i \in \{0, 1\}$ is the event indicator, and $\boldsymbol{x}_i := (x_{ij})_{j \in [p]} \in \mathbb{R}^p$ is a vector composed of p features for each instance $i \in [n]$. We assume that these observations are numbered in the observed order as

$$t_1 \le t_2 \le \cdots \le t_n.$$

The event indicator is defined as

$$\delta_i := \begin{cases} 0 & \text{if the observation is censored} \\ & \text{(i.e., } t_i \text{ is the censoring time),} \\ 1 & \text{otherwise (i.e., } t_i \text{ is the event time)} \end{cases} \qquad (i \in [n]).$$

We assume throughout this paper that the potential censoring time is unrelated to the potential event time.

We introduce the following notation:

$$\boldsymbol{X} := (x_{ij})_{(i,j) \in [n] \times [p]} = \begin{pmatrix} \boldsymbol{x}_1^\top \\ \boldsymbol{x}_2^\top \\ \vdots \\ \boldsymbol{x}_n^\top \end{pmatrix} = \begin{pmatrix} \boldsymbol{x}_{(1)} & \boldsymbol{x}_{(2)} & \cdots & \boldsymbol{x}_{(p)} \end{pmatrix} \in \mathbb{R}^{n \times p}.$$

We then consider the linear regression model

$$\boldsymbol{y} := (y_i)_{i\in[n]} = \boldsymbol{X}\boldsymbol{w} = \left(\boldsymbol{w}^\top \boldsymbol{x}_i\right)_{i\in[n]} \in \mathbb{R}^n,$$

where $\boldsymbol{w} := (w_j)_{j\in[p]} \in \mathbb{R}^p$ is a vector of regression coefficients to be estimated. In the Cox PH model, the instantaneous rate of event occurrence is represented by the *hazard function*, which is defined for time $t \in \mathbb{R}_+$ as

$$h(t \mid \boldsymbol{x}_i) := h_0(t)\exp\left(y_i\right) = h_0(t)\exp\left(\boldsymbol{w}^\top \boldsymbol{x}_i\right) \quad (i \in [n]),$$

where $h_0(t)$ is a baseline hazard function.

The partial likelihood (Cox 1975) of the Cox PH model is then defined as

$$\prod_{i=1}^{n}\left(\frac{h(t_i \mid \boldsymbol{x}_i)}{\sum_{k=i}^{n} h(t_k \mid \boldsymbol{x}_k)}\right)^{\delta_i} = \prod_{i=1}^{n}\left(\frac{\exp\left(\boldsymbol{w}^\top \boldsymbol{x}_i\right)}{\sum_{k=i}^{n}\exp\left(\boldsymbol{w}^\top \boldsymbol{x}_k\right)}\right)^{\delta_i}, \tag{1}$$

which indicates the probability that events will occur in the observed order. In Eq. (1), we assume that there are no ties between event times, whereas some approximation methods (Breslow 1974; Efron 1977) can be applied to the partial likelihood (1) with ties.

The log partial likelihood is expressed as

$$\log\prod_{i=1}^{n}\left(\frac{\exp\left(\boldsymbol{w}^\top \boldsymbol{x}_i\right)}{\sum_{k=i}^{n}\exp\left(\boldsymbol{w}^\top \boldsymbol{x}_k\right)}\right)^{\delta_i} = \sum_{i=1}^{n}\delta_i\left(\boldsymbol{w}^\top \boldsymbol{x}_i - \log\left(\sum_{k=i}^{n}\exp\left(\boldsymbol{w}^\top \boldsymbol{x}_k\right)\right)\right).$$

To estimate the vector $\boldsymbol{w} \in \mathbb{R}^p$ of regression coefficients, we therefore minimize the following loss function with $\boldsymbol{y} = \boldsymbol{X}\boldsymbol{w}$:

$$L(\boldsymbol{y}) := \sum_{i=1}^{n}\delta_i\left(\log\left(\sum_{k=i}^{n}\exp(y_k)\right) - y_i\right), \tag{2}$$

which is known to be convex (see, e.g., Section 3.1.5 in Boyd and Vandenberghe 2004).

To improve generalization performance, we use an L_2-regularization term (Mazumder et al. 2023; Verweij and Van Houwelingen 1994)

$$\frac{1}{2\gamma}\|\boldsymbol{w}\|^2 = \frac{1}{2\gamma}\sum_{j=1}^{p} w_j^2,$$

where $\gamma \in \mathbb{R}_+$ is a user-defined regularization parameter. The L_2-regularized estimation of the Cox PH model is then posed as

$$\underset{\boldsymbol{w}\in\mathbb{R}^p}{\text{minimize}} \quad L(\boldsymbol{X}\boldsymbol{w}) + \frac{1}{2\gamma}\|\boldsymbol{w}\|^2. \tag{3}$$

2.2 Dual formulation of the L_2-regularized estimation

We next derive a dual formulation of problem (3) required for implementing our cutting-plane algorithm. To accomplish this, we begin with the Fenchel conjugate (Boyd and Vandenberghe 2004) of the loss function (2):

$$L^*(\boldsymbol{\alpha}) := \max_{\boldsymbol{y}\in\mathbb{R}^n} \left(\boldsymbol{\alpha}^\top \boldsymbol{y} - L(\boldsymbol{y})\right),$$

where $\boldsymbol{\alpha} := (\alpha_i)_{i\in[n]} \in \mathbb{R}^n$.

Theorem 1 (Wilson et al. 2021) *The Fenchel conjugate of the loss function* (2) *is expressed as*

$$\begin{aligned} L^*(\boldsymbol{\alpha}) = \sum_{i=1}^{n}(\delta_i + \alpha_i)\log(\delta_i + \alpha_i) + \sum_{i=1}^{n-1}\alpha_i \log\left(\frac{\prod_{k=i+1}^{n}\left(\delta_k + \sum_{\ell=k}^{n}\alpha_\ell\right)}{\prod_{k=i+1}^{n}\sum_{\ell=k}^{n}\alpha_\ell}\right) \\ - \sum_{i=1}^{n}\delta_i \log\left(\delta_i + \sum_{k=i}^{n}\alpha_k\right) \end{aligned} \tag{4}$$

with the domain defined as

$$\sum_{i=1}^{n}\alpha_i = 0, \tag{5}$$

$$\sum_{k=i+1}^{n}\alpha_k \geq 0 \quad (i \in [n-1]), \tag{6}$$

$$\delta_i + \alpha_i \geq 0 \quad (i \in [n]). \tag{7}$$

Proof See the supplemental material in Wilson et al. (2021), where the inequalities in Eqs. (6) and (7) are strict. From Theorem 5, however, the Fenchel conjugate function (33) is guaranteed to be bounded by Eqs. (6) and (7), because

$$\begin{aligned} &\zeta_i \log(\zeta_i) = \left(\sum_{k=i}^{n}\alpha_k\right)\log\left(\sum_{k=i}^{n}\alpha_k\right) \to 0 \quad \left(\sum_{k=i}^{n}\alpha_k \searrow 0\right), \\ &(\delta_i + \zeta_i - \zeta_{i+1})\log(\delta_i + \zeta_i - \zeta_{i+1}) = (\delta_i + \alpha_i)\log(\delta_i + \alpha_i) \to 0 \quad (\delta_i + \alpha_i \searrow 0), \end{aligned}$$

where $\boldsymbol{\zeta} := (\zeta_i)_{i\in[n+1]} \in \mathbb{R}^{n+1}$ is defined by Eq. (32). Equations (6) and (7) thus provide a valid domain definition. □

Theorem 2 *Strong duality holds for problem* (3), *and the dual formulation of problem* (3) *is*

$$\underset{\boldsymbol{\alpha}\in\mathbb{R}^n}{\text{maximize}} \quad -L^*(\boldsymbol{\alpha}) - \frac{\gamma}{2}\boldsymbol{\alpha}^\top \boldsymbol{X}\boldsymbol{X}^\top \boldsymbol{\alpha} \tag{8}$$

$$\text{subject to} \quad \sum_{i=1}^{n} \alpha_i = 0, \tag{9}$$

$$\sum_{k=i+1}^{n} \alpha_k \geq 0 \quad (i \in [n-1]), \tag{10}$$

$$\delta_i + \alpha_i \geq 0 \quad (i \in [n]). \tag{11}$$

Proof Problem (3) can then be reformulated as

$$\min_{(\boldsymbol{w},\boldsymbol{y})\in\mathbb{R}^p\times\mathbb{R}^n} \left(L(\boldsymbol{y}) + \frac{1}{2\gamma}\|\boldsymbol{w}\|^2\right) \quad \text{s.t.}\ \boldsymbol{y} = \boldsymbol{X}\boldsymbol{w}. \tag{12}$$

By Slater's condition (Boyd and Vandenberghe 2004), strong duality holds for this problem. The Lagrange dual of problem (12) is formulated as

$$\begin{aligned}
&\min_{(\boldsymbol{w},\boldsymbol{y})\in\mathbb{R}^p\times\mathbb{R}^n} \max_{\boldsymbol{\alpha}\in\mathbb{R}^n} \left(L(\boldsymbol{y}) + \frac{1}{2\gamma}\|\boldsymbol{w}\|^2 - \boldsymbol{\alpha}^\top(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w})\right) \\
&= \max_{\boldsymbol{\alpha}\in\mathbb{R}^n} \left(\min_{\boldsymbol{y}\in\mathbb{R}^n} \left(L(\boldsymbol{y}) - \boldsymbol{\alpha}^\top \boldsymbol{y}\right) + \min_{\boldsymbol{w}\in\mathbb{R}^p} \left(\frac{1}{2\gamma}\|\boldsymbol{w}\|^2 + \boldsymbol{\alpha}^\top \boldsymbol{X}\boldsymbol{w}\right)\right),
\end{aligned}$$

where the first inner minimization problem is converted as

$$\min_{\boldsymbol{y}\in\mathbb{R}^n} \left(L(\boldsymbol{y}) - \boldsymbol{\alpha}^\top \boldsymbol{y}\right) = -\max_{\boldsymbol{y}\in\mathbb{R}^n} \left(\boldsymbol{\alpha}^\top \boldsymbol{y} - L(\boldsymbol{y})\right) = -L^*(\boldsymbol{\alpha}).$$

The optimal solution to the second inner minimization problem is obtained from its optimality condition as

$$\nabla_{\boldsymbol{w}}\left(\frac{1}{2\gamma}\|\boldsymbol{w}\|^2 + \boldsymbol{\alpha}^\top \boldsymbol{X}\boldsymbol{w}\right) = \frac{\boldsymbol{w}}{\gamma} + \boldsymbol{X}^\top \boldsymbol{\alpha} = \boldsymbol{0} \quad \Rightarrow \quad \boldsymbol{w}^\star = -\gamma \boldsymbol{X}^\top \boldsymbol{\alpha},$$

reducing the second inner minimization problem to

$$\frac{1}{2\gamma}\|\boldsymbol{w}^\star\|^2 + \boldsymbol{\alpha}^\top \boldsymbol{X}\boldsymbol{w}^\star = -\frac{\gamma}{2}\boldsymbol{\alpha}^\top \boldsymbol{X}\boldsymbol{X}^\top \boldsymbol{\alpha}.$$

Imposing domain constraints (5)–(7) on the Lagrange dual problem completes the proof. □

2.3 Bilevel optimization formulation for sparse estimation

Let $\boldsymbol{z} := (z_j)_{j \in [p]} \in \{0,1\}^p$ be a vector composed of binary decision variables for subset selection; namely $z_j = 1$ if the jth feature is selected, and $z_j = 0$ otherwise. We pose estimation of sparse Cox PH models as

$$\underset{(\boldsymbol{w},\boldsymbol{z}) \in \mathbb{R}^p \times \{0,1\}^p}{\text{minimize}} \quad L(\boldsymbol{X}\boldsymbol{w}) + \frac{1}{2\gamma}\|\boldsymbol{w}\|^2 \tag{13}$$

$$\text{subject to} \quad z_j = 0 \ \Rightarrow \ w_j = 0 \quad (j \in [p]), \tag{14}$$

$$\sum_{j=1}^{p} z_j = \theta, \tag{15}$$

where $\theta \in [p]$ is a user-defined parameter for specifying the subset size through constraint (15). If $z_j = 0$, then the jth regression coefficient must be zero due to the logical implication (14), which can be imposed using indicator constraints implemented in modern optimization software. The logical implication (14) can also be represented as

$$-Mz_j \le w_j \le Mz_j \quad (j \in [p]),$$

where $M \in \mathbb{R}_+$ is a sufficiently large positive constant.

It is very difficult to handle problem (13)–(15), which is a mixed-integer nonlinear optimization problem. Following Bertsimas et al. (2021), we thus consider a reformulation of our sparse estimation into the bilevel optimization to separate problem (13)–(15) into discrete and continuous optimization problems. Specifically, the upper-level problem for subset selection is written as the integer optimization problem

$$\underset{\boldsymbol{z} \in \{0,1\}^p}{\text{minimize}} \quad f(\boldsymbol{z}) \tag{16}$$

$$\text{subject to} \quad \sum_{j=1}^{p} z_j = \theta, \tag{17}$$

and the lower-level problem for calculating the objective function is expressed as the nonlinear convex optimization problem

$$f(\boldsymbol{z}) = \underset{\boldsymbol{w} \in \mathbb{R}^p}{\text{minimize}} \quad L(\boldsymbol{X}\boldsymbol{w}) + \frac{1}{2\gamma}\|\boldsymbol{w}\|^2 \tag{18}$$

$$\text{subject to} \quad z_j = 0 \ \Rightarrow \ w_j = 0 \quad (j \in [p]). \tag{19}$$

Theorem 3 *For $\boldsymbol{z} \in \{0,1\}^p$, the dual formulation of problem* (18) *and* (19) *becomes*

$$f(z) = \underset{\boldsymbol{\alpha} \in \mathbb{R}^n}{\text{maximize}} \quad -L^*(\boldsymbol{\alpha}) - \frac{\gamma}{2} \sum_{j=1}^{p} z_j \boldsymbol{\alpha}^\top \boldsymbol{x}_{(j)} \boldsymbol{x}_{(j)}^\top \boldsymbol{\alpha} \tag{20}$$

$$\text{subject to} \quad \sum_{i=1}^{n} \alpha_i = 0, \tag{21}$$

$$\sum_{k=i+1}^{n} \alpha_k \geq 0 \quad (i \in [n-1]), \tag{22}$$

$$\delta_i + \alpha_i \geq 0 \quad (i \in [n]). \tag{23}$$

Proof According to $z \in \{0,1\}^p$, we define the submatrix of features as

$$\boldsymbol{X}_z := \left(\boldsymbol{x}_{(j)} \mid j \in [p],\ z_j = 1\right) \in \mathbb{R}^{n \times p(z)},$$

where $p(z) := \sum_{j=1}^{p} z_j$. Then, the lower-level problem (18) and (19) can be rewritten as

$$\underset{\boldsymbol{w} \in \mathbb{R}^{p(z)}}{\text{minimize}} \quad L(\boldsymbol{X}_z \boldsymbol{w}) + \frac{1}{2\gamma} \|\boldsymbol{w}\|^2. \tag{24}$$

From Theorem 2, the associated dual formulation is represented as

$$\underset{\boldsymbol{\alpha} \in \mathbb{R}^n}{\text{maximize}} \quad -L^*(\boldsymbol{\alpha}) - \frac{\gamma}{2} \boldsymbol{\alpha}^\top \boldsymbol{X}_z \boldsymbol{X}_z^\top \boldsymbol{\alpha} \tag{25}$$

$$\text{subject to} \quad \text{Eqs. (21)–(23).} \tag{26}$$

Note here that

$$\boldsymbol{\alpha}^\top \boldsymbol{X}_z \boldsymbol{X}_z^\top \boldsymbol{\alpha} = \sum_{j=1}^{p} z_j \boldsymbol{\alpha}^\top \boldsymbol{x}_{(j)} \boldsymbol{x}_{(j)}^\top \boldsymbol{\alpha},$$

which completes the proof. □

Theorem 3 allows us to redefine $f(z)$ as the optimal objective value of problem (20)–(23) for real-valued vector $z \in [0,1]^p$. In this case, following Bertsimas et al. (2021), we can see that $f(z)$ is a convex function with a subgradient given by

$$\boldsymbol{g}(z) := -\frac{\gamma}{2} \left(\boldsymbol{\alpha}^\star(z)^\top \boldsymbol{x}_{(j)} \boldsymbol{x}_{(j)}^\top \boldsymbol{\alpha}^\star(z)\right)_{j \in [p]} \in \partial f(z) \subseteq \mathbb{R}^p, \tag{27}$$

where $\boldsymbol{\alpha}^\star(z) \in \mathbb{R}^n$ is an optimal solution to the dual lower-level problem (20)–(23).

3 Cutting-plane algorithm

This section describes our cutting-plane algorithm for estimation of sparse Cox PH models. To accelerate the cutting-plane algorithm, we also derive a quadratic approximation of the Fenchel conjugate function.

3.1 Algorithm description

We now extend the cutting-plane algorithm (Bertsimas et al. 2021) to estimation of sparse Cox PH models. Our algorithm, which is based on a reformulation of our sparse estimation into bilevel optimization, aims to solve the upper-level problem (16) and (17).

Let $\xi_{\mathrm{LB}} \in \mathbb{R}$ be a lower bound on the optimal objective value $f^\star$ of problem (16) and (17) (i.e., $\xi_{\mathrm{LB}} \leq f^\star$). For example, this bound can be calculated by solving problem (3). Our cutting-plane algorithm starts with the initial feasible region

$$\mathcal{F}_1 := \left\{ (z, \xi) \in \{0,1\}^p \times \mathbb{R} \;\middle|\; \sum_{j=1}^{p} z_j = \theta, \quad \xi \geq \xi_{\mathrm{LB}} \right\}, \tag{28}$$

where $\xi \in \mathbb{R}$ is an auxiliary decision variable that corresponds to a lower estimate of $f(z)$.

At the kth iteration ($k \geq 1$), our algorithm solves a surrogate version of the upper-level problem (16) and (17)

$$\underset{(z,\xi) \in \{0,1\}^p \times \mathbb{R}}{\text{minimize}} \quad \xi \tag{29}$$

$$\text{subject to} \quad (z, \xi) \in \mathcal{F}_k, \tag{30}$$

where $\mathcal{F}_k$ is a feasible region at the kth iteration such that $\mathcal{F}_k \subseteq \mathcal{F}_1$. Because the objective value is bounded below by Eq. (28), there exists an optimal solution $(z^{(k)}, \xi^{(k)})$ to problem (29) and (30).

We next solve the dual lower-level problem (20)–(23) with $z = z^{(k)}$, thus obtaining the objective value $f(z^{(k)})$ and its subgradient $\boldsymbol{g}(z^{(k)})$ from Eq. (27). If $f(z^{(k)}) \leq \xi^{(k)} + \varepsilon$ holds with sufficiently small $\varepsilon \geq 0$, then $z^{(k)}$ is an ε-optimal solution to problem (16) and (17), namely

$$f(z^{(k)}) \leq f^\star + \varepsilon.$$

In this case, we terminate the algorithm with the ε-optimal solution $z^{(k)}$. Otherwise, we add a linear underestimator of $f(z)$ to the set of constraints:

$$\mathcal{F}_{k+1} \leftarrow \mathcal{F}_k \cap \{(z, \xi) \in \{0,1\}^p \times \mathbb{R} \mid \xi \geq f(z^{(k)}) + \boldsymbol{g}(z^{(k)})^\top (z - z^{(k)})\}. \tag{31}$$

Note that because $\xi^{(k)} < f(z^{(k)})$, this update cuts off the solution $(z^{(k)}, \xi^{(k)})$.

We set $k \leftarrow k+1$ and then use the refined feasible region (31) to again solve the surrogate upper-level problem (29) and (30). We repeat this procedure until we find an ε-optimal solution $\hat{z}$.

Algorithm 1 summarizes our cutting-plane algorithm. Note that the surrogate upper-level problem (29) and (30) is a mixed-integer linear optimization problem, which can be solved to optimality using optimization software. Following Kobayashi et al. (2021, 2023), we can prove the finite convergence of the algorithm.

Theorem 4 (Kobayashi et al. 2021, 2023) *Algorithm* 1 *terminates in a finite number of iterations and outputs an ε-optimal solution to problem* (16) *and* (17).

Proof See Kobayashi et al. (2021, 2023). □

Algorithm 1 Cutting-plane algorithm for solving problem (16)–(17)

Step 0 (*Initialization*): Let $\varepsilon \geq 0$ be the tolerance for optimality. Define the feasible region $\mathcal{F}_1$ as in Eq. (28). Set $k \leftarrow 1$ and $\mathrm{UB}_0 \leftarrow \infty$.
Step 1 (*Surrogate upper-level problem*): Solve problem (29)–(30). Let $(\boldsymbol{z}^{(k)}, \xi^{(k)})$ be an optimal solution, and set $\mathrm{LB}_k \leftarrow \xi^{(k)}$.
Step 2 (*Dual lower-level problem*): Solve problem (20)–(23) with $\boldsymbol{z} = \boldsymbol{z}^{(k)}$ to obtain $f(\boldsymbol{z}^{(k)})$ and calculate $\boldsymbol{g}(\boldsymbol{z}^{(k)})$ as in Eq. (27). If $f(\boldsymbol{z}^{(k)}) < \mathrm{UB}_{k-1}$, set $\mathrm{UB}_k \leftarrow f(\boldsymbol{z}^{(k)})$ and $\hat{\boldsymbol{z}} \leftarrow \boldsymbol{z}^{(k)}$; otherwise, set $\mathrm{UB}_k \leftarrow \mathrm{UB}_{k-1}$.
Step 3 (*Termination condition*): If $\mathrm{UB}_k - \mathrm{LB}_k \leq \varepsilon$, terminate the algorithm with the ε-optimal solution $\hat{\boldsymbol{z}}$.
Step 4 (*Cut generation*): Update the feasible region as in Eq. (31). Set $k \leftarrow k+1$ and return to Step 1.

3.2 Quadratic approximation of the Fenchel conjugate function

Note that Step 2 of Algorithm 1 solves the nonlinear convex optimization problem (20)–(23) in every iteration to generate cutting planes. To accelerate this computation, we use a quadratic approximation of the Fenchel conjugate function (4) in the objective function (20).

Since the accuracy of quadratic approximation is generally low for multivariate functions, we rewrite the multivariate conjugate function (4) as the sum of univariate functions, each of which can be approximated accurately by a quadratic function. For notational simplicity, we introduce $\boldsymbol{\zeta} := (\zeta_i)_{i \in [n+1]} \in \mathbb{R}^{n+1}$ as

$$\zeta_i := \sum_{k=i}^{n} \alpha_k \quad (i \in [n]), \qquad \zeta_{n+1} := 0. \tag{32}$$

The following theorem aids in converting the multivariate conjugate function (4) into the sum of univariate functions.

Theorem 5 *The Fenchel conjugate function* (4) *is represented by* $\boldsymbol{\zeta} \in \mathbb{R}^{n+1}$ *through Eq.* (32) *as*

$$\begin{aligned} L^*(\boldsymbol{\alpha}) = \sum_{i=1}^{n} (\delta_i + \zeta_i - \zeta_{i+1}) \log(\delta_i + \zeta_i - \zeta_{i+1}) \\ + \sum_{i=1}^{n} \zeta_i \log(\zeta_i) - \sum_{i=1}^{n} (\delta_i + \zeta_i) \log(\delta_i + \zeta_i) \end{aligned} \tag{33}$$

with the domain defined as

$$\zeta_1 = 0, \tag{34}$$

$$\zeta_{i+1} \geq 0 \quad (i \in [n-1]), \tag{35}$$

$$\delta_i + \zeta_i - \zeta_{i+1} \geq 0 \quad (i \in [n]). \tag{36}$$

Proof See Appendix 1. □

From Theorem 5, we can express the Fenchel conjugate function (4) as the sum of univariate functions

$$\begin{aligned} L^*(\boldsymbol{\alpha}) &= \sum_{i=1}^{n} (\delta_i + \alpha_i) \log(\delta_i + \alpha_i) + \sum_{i=1}^{n} \zeta_i \log(\zeta_i) - \sum_{i=1}^{n} (\delta_i + \zeta_i) \log(\delta_i + \zeta_i) \\ &= \sum_{i=1}^{n} f_1(\alpha_i \mid \delta_i) + \sum_{i=1}^{n} f_2(\zeta_i \mid \delta_i), \end{aligned}$$

where

$$f_1(\alpha \mid \delta) := (\delta + \alpha) \log(\delta + \alpha), \tag{37}$$

$$f_2(\zeta \mid \delta) := \zeta \log(\zeta) - (\delta + \zeta) \log(\delta + \zeta) \tag{38}$$

for $\delta \in \{0, 1\}$. These univariate convex functions can be approximated accurately by quadratic functions as

$$f_1(\alpha \mid \delta) \approx \tilde{f}_1(\alpha \mid \delta) := q_{11}^{(\delta)} \alpha^2 + q_{12}^{(\delta)} \alpha + q_{13}^{(\delta)}, \tag{39}$$

$$f_2(\zeta \mid \delta) \approx \tilde{f}_2(\zeta \mid \delta) := q_{21}^{(\delta)} \zeta^2 + q_{22}^{(\delta)} \zeta + q_{23}^{(\delta)}, \tag{40}$$

where $\boldsymbol{q}_1^{(\delta)} := (q_{11}^{(\delta)}, q_{12}^{(\delta)}, q_{13}^{(\delta)})^\top \in \mathbb{R}^3$ and $\boldsymbol{q}_2^{(\delta)} := (q_{21}^{(\delta)}, q_{22}^{(\delta)}, q_{23}^{(\delta)})^\top \in \mathbb{R}^3$ are coefficient vectors of quadratic functions for $\delta \in \{0, 1\}$. Accordingly, we obtain a quadratic approximation of the Fenchel conjugate function (4) as

$$\begin{aligned} L^*(\boldsymbol{\alpha}) \approx \tilde{L}^*(\boldsymbol{\alpha}) &:= \sum_{i=1}^{n} \tilde{f}_1(\alpha_i \mid \delta_i) + \sum_{i=1}^{n} \tilde{f}_2(\zeta_i \mid \delta_i) \\ &= \sum_{i=1}^{n} \tilde{f}_1(\alpha_i \mid \delta_i) + \sum_{i=1}^{n} \tilde{f}_2\left(\sum_{k=i}^{n} \alpha_k \;\middle|\; \delta_i \right). \quad \because \text{Eq. (32)} \end{aligned}$$

Consequently, the nonlinear convex optimization problem (20)–(23) is reduced to the quadratic optimization problem

$$f(\boldsymbol{z}) \approx \underset{\boldsymbol{\alpha} \in \mathbb{R}^n}{\text{maximize}} \quad -\tilde{L}^*(\boldsymbol{\alpha}) - \frac{\gamma}{2} \sum_{j=1}^{p} z_j \boldsymbol{\alpha}^\top \boldsymbol{x}_{(j)} \boldsymbol{x}_{(j)}^\top \boldsymbol{\alpha} \tag{41}$$

$$\text{subject to} \quad \text{Eqs. (21)–(23).} \tag{42}$$

Accordingly, we can revise Step 2 of Algorithm 1 as follows:

Step 2 (*Approximate dual lower-level problem*): Solve problem (41) and (42) with $\boldsymbol{z} = \boldsymbol{z}^{(k)}$ to obtain $f(\boldsymbol{z}^{(k)})$ and calculate $\boldsymbol{g}(\boldsymbol{z}^{(k)})$ as in Eq. (27). If $f(\boldsymbol{z}^{(k)}) < \mathrm{UB}_{k-1}$, set $\mathrm{UB}_k \leftarrow f(\boldsymbol{z}^{(k)})$ and $\hat{\boldsymbol{z}} \leftarrow \boldsymbol{z}^{(k)}$; otherwise, set $\mathrm{UB}_k \leftarrow \mathrm{UB}_{k-1}$.

3.3 Least-squares method for quadratic approximation

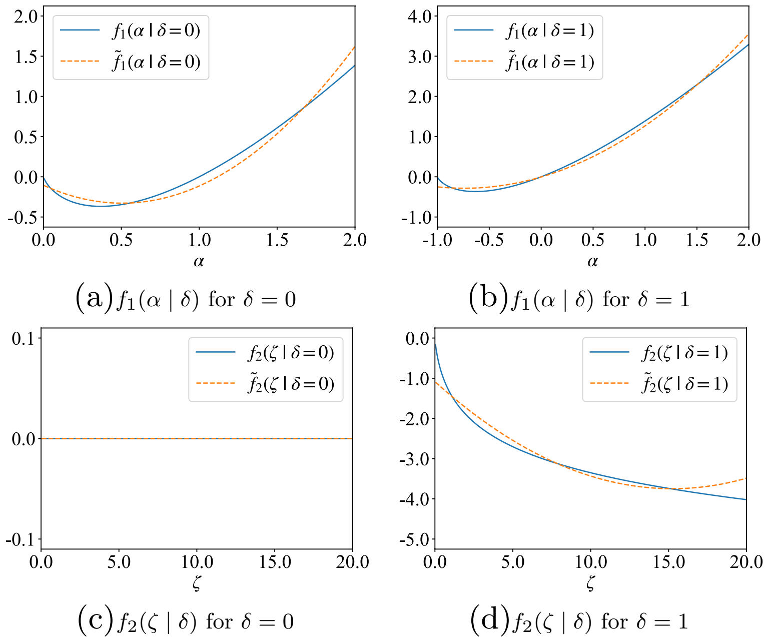
We implement a computationally efficient method for determining appropriate values of coefficients (i.e., $\boldsymbol{q}_1^{(\delta)}$ and $\boldsymbol{q}_2^{(\delta)}$ for $\delta \in \{0, 1\}$) of quadratic functions (39) and (40) for each dataset. Let $\boldsymbol{\alpha}^\star := (\alpha_i^\star)_{i \in [n]} \in \mathbb{R}^n$ be an optimal solution to the dual lower-level problem (8)–(11), which corresponds to the full model (3) without subset selection and can be solved using nonlinear optimization software. According to Eq. (32), we define $\boldsymbol{\zeta}^\star := (\zeta_i^\star)_{i \in [n+1]} \in \mathbb{R}^{n+1}$ based on $\boldsymbol{\alpha}^\star$. We then solve the following least-squares problems for minimizing the sum of squared approximation gaps based on $\boldsymbol{\alpha}^\star$ and $\boldsymbol{\zeta}^\star$:

$$\underset{\boldsymbol{q}_1^{(\delta)} \in \mathbb{R}^3}{\text{minimize}} \quad \sum_{i \in \mathcal{N}_\delta} \left(f_1(\alpha_i^\star \mid \delta_i) - \left(q_{11}^{(\delta)} (\alpha_i^\star)^2 + q_{12}^{(\delta)} \alpha_i^\star + q_{13}^{(\delta)} \right) \right)^2, \tag{43}$$

$$\underset{\boldsymbol{q}_2^{(\delta)} \in \mathbb{R}^3}{\text{minimize}} \quad \sum_{i \in \mathcal{N}_\delta} \left(f_2\left(\zeta_i^\star \mid \delta_i\right) - \left(q_{21}^{(\delta)} (\zeta_i^\star)^2 + q_{22}^{(\delta)} \zeta_i^\star + q_{23}^{(\delta)} \right) \right)^2, \tag{44}$$

where $\mathcal{N}_\delta := \{ i \in [n] \mid \delta_i = \delta \}$ for $\delta \in \{0, 1\}$. It is well known that such least-squares problems can be solved analytically (Boyd and Vandenberghe 2004).

Figure 1 shows examples of quadratic functions (dashed curves) for approximating the univariate convex functions (37) and (38) (solid curves). Here, quadratic approximations were computed using the least-squares method (43) and (44) for a synthetic dataset ($n = 100,\ p = 10,\ \theta^\star = 5,\ \mathrm{SNR} = 10^0$); see Sect. 4.3 for details of the dataset.



(a) $f_1(\alpha \mid \delta)$ for $\delta = 0$

(b) $f_1(\alpha \mid \delta)$ for $\delta = 1$

(c) $f_2(\zeta \mid \delta)$ for $\delta = 0$

(d) $f_2(\zeta \mid \delta)$ for $\delta = 1$

Fig. 1 Examples of quadratic approximations of $f_1(\alpha \mid \delta)$ and $f_2(\zeta \mid \delta)$

4 Computational results

This section evaluates the effectiveness of our method for sparse estimation through computational experiments using synthetic and real-world datasets. All computations were performed on a Windows computer with an Intel Core i7-10700 CPU (2.90 GHz) and 16 GB of memory.

4.1 Methods for comparison

We compare the performance of the following methods for estimation of sparse Cox PH models:

CPA: The cutting-plane algorithm (Algorithm 1);

CPA+: The cutting-plane algorithm (Algorithm 1) using the quadratic approximation problem (41) and (42), where quadratic functions (39) and (40) were determined by the least-squares method (43) and (44);

Exhst: Exhaustive search method, which performs the L_2-regularized estimation (3) for all possible subsets of features;

L1Rgl: L_1-regularized estimation (Tibshirani 1997);

ENet: Elastic-net-regularized estimation (Park and Hastie 2007).

We implemented these methods in Python. In the cutting-plane algorithms, the mixed-integer linear optimization problem (29) and (30) and convex quadratic optimization problem (41) and (42) were solved using the optimization software Gurobi Optimizer 9.5.0,[1] and the nonlinear convex optimization problem (20)–(23) was solved using the optimization software Ipopt 3.1.1[2] (Wächter and Biegler 2006). We used the lazy constraint callback to add linear constraints (31) during a branch-and-bound procedure. We set $\varepsilon = 10^{-2}$ as the tolerance for optimality and selected $\gamma \in \{10^{-2}, 10^{-1}, \dots, 10^{2}\}$ as the L_2-regularization parameter. We implemented regularized estimation methods using the Python `lifelines` library[3] (Davidson-Pilon 2019), where the `penalizer` and `l1_ratio` parameters for regularization were chosen from $\{10^{-2}, 10^{-1}, \dots, 10^{2}\}$ and $\{0.0, 0.1, \dots, 1.0\}$, respectively.

4.2 Performance evaluation methodology

We partitioned a whole set of data instances into training and testing datasets. Using a training dataset, we selected a subset $\hat{\mathcal{S}}$ of features and then trained the L_2-regularized Cox PH model (3) with the selected features, thereby obtaining a vector $\hat{\boldsymbol{w}} := (\hat{w}_j)_{j \in [p]} \in \mathbb{R}^p$ of regression coefficients such that $\hat{w}_j = 0$ for $j \notin \hat{\mathcal{S}}$. Here, regularization parameters (i.e., γ, `penalizer`, and `l1_ratio`) were tuned through hold-out validation using the training dataset. After that, we evaluated the out-of-sample prediction accuracy by applying the trained Cox PH model to a testing dataset.

The accuracy of subset selection is measured by the true positive (TP) and false positive (FP) rates, defined as

$$\textbf{TP rate} := \frac{|\mathcal{S}^\star \cap \hat{\mathcal{S}}|}{|\mathcal{S}^\star|}, \qquad \textbf{FP rate} := \frac{|([p] \setminus \mathcal{S}^\star) \cap \hat{\mathcal{S}}|}{|[p] \setminus \mathcal{S}^\star|},$$

where $\mathcal{S}^\star$ is the index set of relevant features, given for synthetic datasets as in Eq. (45). The out-of-sample prediction accuracy is quantified by the concordance index (C-index) (Harrell et al. 1996; Uno et al. 2011), a rank-correlation measure often used for censored survival data. The C-index is defined as

[1] https://www.gurobi.com.

[2] https://github.com/coin-or/Ipopt.

[3] https://lifelines.readthedocs.io/en/latest/.

$$\textbf{C-index} := \frac{\sum_{i=1}^{n}\sum_{k=1}^{n}\delta_i \mathbf{1}(t_i < t_k)\mathbf{1}(\hat{\boldsymbol{w}}^\top \boldsymbol{x}_i > \hat{\boldsymbol{w}}^\top \boldsymbol{x}_k)}{\sum_{i=1}^{n}\sum_{k=1}^{n}\delta_i \mathbf{1}(t_i < t_k)},$$

where $\mathbf{1}(\,\cdot\,)$ is the indicator function; namely, $\mathbf{1}(P) = 1$ if the proposition P is true, and $\mathbf{1}(P) = 0$ otherwise.

We assess the computational efficiency of subset selection by

#(Iter): the number of iterations of the cutting-plane algorithms, and

Time: the computation time in seconds required for subset selection and hold-out validation of regularization parameters.

4.3 Generation of synthetic datasets

In a similar manner to Simon et al. (2011), we prepared synthetic datasets according to the following steps, with $\theta^\star := 5$ set as the true subset size. First, we defined a vector $\boldsymbol{w}^\star := (w_j^\star)_{j\in[p]} \in \mathbb{R}^p$ of true coefficients as

$$w_j^\star := \begin{cases} (-0.8)^{j-1} & \text{if } j \in [\theta^\star], \\ 0 & \text{otherwise,} \end{cases} \qquad \mathcal{S}^\star := [\theta^\star]. \tag{45}$$

Next, we sampled feature vectors from a multivariate normal distribution as $\boldsymbol{x}_i \sim \mathrm{N}(\mathbf{0}, \boldsymbol{\Sigma})$, where $\boldsymbol{\Sigma} := (\sigma_{ij})_{(i,j)\in[p]\times[p]} \in \mathbb{R}^{p\times p}$ is the covariance matrix with $\sigma_{ij} := 0.5^{|i-j|}$. We then generated true event times based on the signal-to-noise ratio (SNR) and a normal random number $z_i^{(1)} \sim \mathrm{N}(0, 1)$ as

$$e_i := \exp\left((\boldsymbol{w}^\star)^\top \boldsymbol{x}_i + \kappa z_i^{(1)}\right) \quad (i \in [n]),$$

where

$$\kappa := \sqrt{\frac{(\boldsymbol{w}^\star)^\top \boldsymbol{\Sigma} \boldsymbol{w}^\star}{\mathrm{SNR}}}.$$

Note that when the SNR is high, the random term $\kappa z_i^{(1)}$ has a small impact on the event time. Similarly, we generated censoring times based on a normal random number $z_i^{(2)} \sim \mathrm{N}(0, 1)$ as

$$c_i := \exp(\kappa z_i^{(2)}) \quad (i \in [n]).$$

Finally, we set the observed time and the event indicator as

Table 1 Results for the synthetic low-dimensional datasets ($n = 200,\ p = 20,\ \theta = \theta^\star = 5$)

SNR	Method	C-index	TP rate	FP rate	$\lvert\hat{S}\rvert$	#(Iter)	Time
10^{-1}	CPA	0.569 (± 0.008)	0.52 (± 0.07)	0.16 (± 0.02)	5.0	2649.5	336.0
	CPA+	**0.578** (± 0.006)	0.64 (± 0.07)	**0.12** (± 0.02)	5.0	3184.0	47.8
	L1Rgl(θ)	0.571 (± 0.006)	0.44 (± 0.05)	0.19 (± 0.02)	5.0	–	3.7
	ENet(θ)	0.567 (± 0.006)	0.38 (± 0.04)	0.21 (± 0.02)	5.0	–	20.8
	L1Rgl(hv)	0.576 (± 0.007)	0.52 (± 0.15)	0.47 (± 0.15)	9.6	–	1.4
	ENet(hv)	0.574 (± 0.007)	**0.84** (± 0.11)	0.71 (± 0.13)	14.8	–	11.9
10^{0}	CPA	0.771 (± 0.006)	0.92 (± 0.04)	0.03 (± 0.01)	5.0	1473.7	181.9
	CPA+	**0.773** (± 0.005)	**0.94** (± 0.03)	**0.02** (± 0.01)	5.0	1732.2	26.6
	L1Rgl(θ)	0.744 (± 0.006)	0.76 (± 0.03)	0.08 (± 0.01)	5.0	–	3.5
	ENet(θ)	0.737 (± 0.008)	0.72 (± 0.04)	0.09 (± 0.01)	5.0	–	23.7
	L1Rgl(hv)	0.754 (± 0.009)	0.88 (± 0.06)	0.65 (± 0.13)	14.3	–	1.5
	ENet(hv)	0.753 (± 0.007)	0.88 (± 0.08)	0.45 (± 0.11)	11.4	–	11.9
10^{1}	CPA	**0.929** (± 0.000)	**1.00** (± 0.00)	**0.00** (± 0.00)	5.0	1065.9	131.4
	CPA+	**0.929** (± 0.000)	**1.00** (± 0.00)	**0.00** (± 0.00)	5.0	1167.2	18.8
	L1Rgl(θ)	0.924 (± 0.005)	0.98 (± 0.02)	0.01 (± 0.01)	5.0	–	2.7
	ENet(θ)	0.899 (± 0.014)	0.90 (± 0.04)	0.03 (± 0.01)	5.0	–	19.1
	L1Rgl(hv)	0.924 (± 0.001)	**1.00** (± 0.00)	0.98 (± 0.01)	18.9	–	1.4
	ENet(hv)	0.920 (± 0.003)	**1.00** (± 0.00)	0.83 (± 0.10)	17.0	–	12.0

$$t_i := \min\{e_i, c_i\} \quad (i \in [n]),$$

$$\delta_i := \begin{cases} 0 & \text{if } c_i \le e_i, \\ 1 & \text{otherwise} \end{cases} \quad (i \in [n]).$$

We used n data instances as the training dataset and generated sufficiently many data instances for the testing datasets.

4.4 Results for synthetic datasets

Tables 1 and 2 give the computational results with the subset size parameter $\theta = \theta^\star = 5$ and SNR $\in \{10^{-1}, 10^0, 10^1\}$ for the synthetic datasets. In these experiments, we terminated a computation of the cutting-plane algorithms if it did not complete within 100 s; in those cases, the best feasible solution found within 100 s was taken as the result. We repeated data generation and performance evaluation 10 times and calculated mean values. In the tables, values in parentheses are standard errors for the C-index, TP rate, and FP rate. The largest C-index and TP rate values and the smallest FP rate values for each problem instance are shown in bold. Note that L1Rgl(hv) and ENet(hv) tuned regularization parameters through the hold-out validation with the training dataset, so they often selected more than θ features. In addition, L1Rgl(θ) and ENet(θ) selected θ features by tuning the regularization

Table 2 Results for the synthetic small-sample datasets ($n = 50,\ p = 80,\ \theta = \theta^{\star} = 5$)

SNR	Method	C-index	TP rate	FP rate	$\lvert\hat{S}\rvert$	#(Iter)	Time
10^{-1}	CPA	0.501 (± 0.003)	0.06 (± 0.03)	**0.06** (± 0.00)	5.0	3117.6	363.1
	CPA+	0.501 (± 0.004)	0.06 (± 0.03)	**0.06** (± 0.00)	5.0	3279.8	63.1
	L1Rgl(θ)	0.498 (± 0.004)	0.04 (± 0.03)	**0.06** (± 0.00)	5.0	–	5.8
	ENet(θ)	0.499 (± 0.005)	0.04 (± 0.03)	**0.06** (± 0.00)	5.0	–	34.8
	L1Rgl(hv)	**0.511** (± 0.004)	0.28 (± 0.09)	0.17 (± 0.05)	14.1	–	1.7
	ENet(hv)	0.506 (± 0.005)	**0.54** (± 0.14)	0.50 (± 0.13)	40.0	–	14.7
10^{0}	CPA	0.601 (± 0.019)	0.30 (± 0.05)	**0.05** (± 0.00)	5.0	3026.2	331.0
	CPA+	0.590 (± 0.019)	0.24 (± 0.06)	**0.05** (± 0.00)	5.0	3131.4	62.5
	L1Rgl(θ)	0.599 (± 0.018)	0.28 (± 0.06)	**0.05** (± 0.00)	5.0	–	8.3
	ENet(θ)	0.580 (± 0.016)	0.28 (± 0.06)	**0.05** (± 0.00)	5.0	–	38.6
	L1Rgl(hv)	0.594 (± 0.014)	0.36 (± 0.11)	0.12 (± 0.03)	11.0	–	1.7
	EN(hv)	**0.609** (± 0.017)	**0.58** (± 0.10)	0.30 (± 0.10)	25.1	–	14.8
10^{1}	CPA	**0.856** (± 0.027)	0.76 (± 0.10)	**0.02** (± 0.01)	5.0	3005.6	332.3
	CPA+	0.841 (± 0.023)	0.72 (± 0.10)	**0.02** (± 0.01)	5.0	3060.8	60.0
	L1Rgl(θ)	0.746 (± 0.013)	0.42 (± 0.04)	0.04 (± 0.00)	5.0	–	7.3
	ENet(θ)	0.713 (± 0.012)	0.34 (± 0.06)	0.04 (± 0.00)	5.0	–	38.3
	L1Rgl(hv)	0.793 (± 0.009)	**0.80** (± 0.04)	0.24 (± 0.05)	21.9	–	1.7
	ENet(hv)	0.750 (± 0.010)	0.40 (± 0.07)	0.06 (± 0.03)	6.5	–	14.8

parameters such that the number of features with nonzero regression coefficients was θ.

We first focus on the results for low-dimensional datasets ($n = 200,\ p = 20$) in Table 1. Our cutting-plane algorithm with the quadratic approximation (CPA+) delivered the overall best performance in terms of accuracy for both prediction (C-index) and subset selection (TP and FP rates). The cutting-plane algorithm without the quadratic approximation (CPA) required much longer computation times than CPA+ did. Although the regularized estimation methods (L1Rgl and ENet) were fast, their performances of prediction and subset selection were often worse than with the cutting-plane algorithms; in particular, the regularized estimation methods with the hold-out validation had very large FP rates.

We next move on to the results for small-sample datasets ($n = 50,\ p = 80$) in Table 2. When the SNR was low, the regularized estimation methods with the hold-out validation attained high C-index values. A main reason for this is that they selected many features, which worsened the FP rate but increased the TP rate. In contrast, when the SNR was high, our cutting-plane algorithms performed very well in terms of accuracy for both prediction and subset selection. Our cutting-plane algorithm was still much faster with the quadratic approximation than without it.

Figure 2 shows the performance of our cutting-plane algorithms (CPA and CPA+) and the exhaustive search method (Exhst) with the number of candidate features $p \in \{6, 8, 10, 12, 14\}$ for relatively small synthetic datasets ($n = 100,\ \theta = \theta^{\star} = 5,\ \mathrm{SNR} = 10^{0},\ \gamma = 10^{2}$). In these figures, mean values of performance metrics are shown with standard errors represented by error bars. The

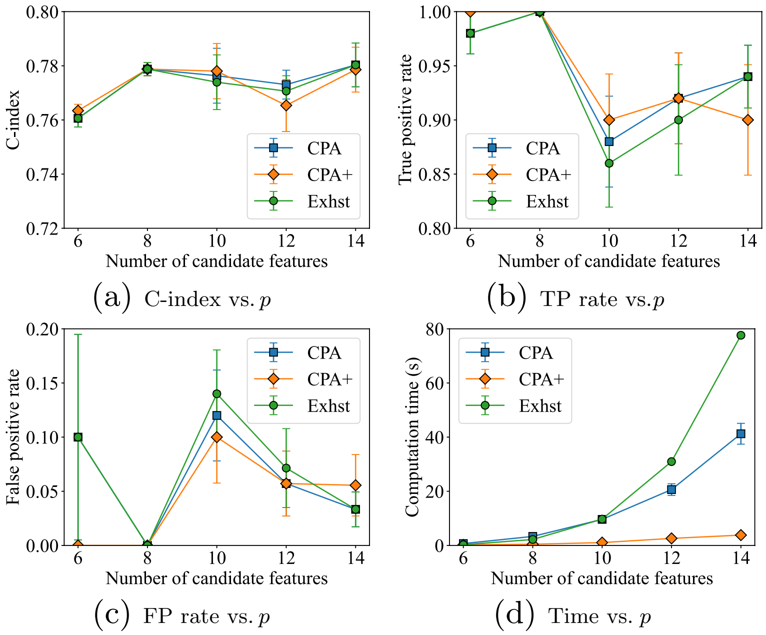
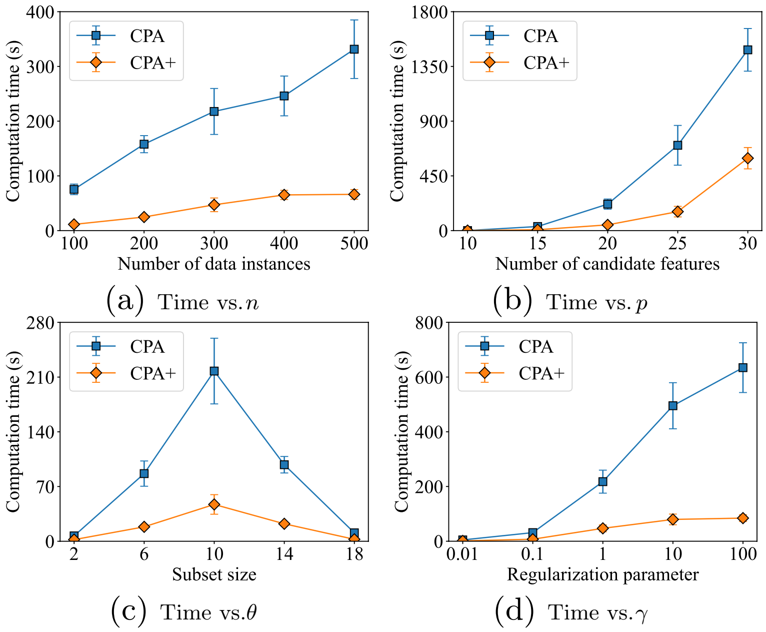


Fig. 2 Performance comparison with the exhaustive search method for synthetic datasets ($n = 100,\ \theta = \theta^\star = 5,\ \mathrm{SNR} = 10^0,\ \gamma = 10^2$)

three algorithms delivered similar performance in terms of accuracy for both prediction and subset selection (Fig. 2a–c). On the other hand, the computation time required by the exhaustive search method increased sharply with the number of candidate features (Fig. 2d).

Figure 3 shows the computation time of our cutting-plane algorithms (CPA and CPA+) with various parameter values for synthetic datasets ($\mathrm{SNR} = 10^0,\ \theta^\star = 10$), where the default parameter values are $n = 300,\ p = 20,\ \theta = 10,\ \gamma = 10^0$. The computation time increased slowly with the number of data instances (Fig. 3a) and rapidly with the number of candidate features (Fig. 3b). The computation time was largest when the subset size was half the number of candidate features (i.e., $\theta = p/2 = 10$) (Fig. 3c). The difference in the computation time between the algorithms was very large when the regularization parameter was set to large values (Fig. 3d).

Figure 4 shows the performance of our cutting-plane algorithm (CPA+) and regularized estimation methods (L1Rgl(θ) and ENet(θ)) as a function of the number of data instances for synthetic datasets ($p = 20,\ \theta = \theta^\star = 5$). When $\mathrm{SNR} = 10^{-1}$, the three methods similarly improved all the metrics with the increasing number of data instances (Fig. 4a–c). When $\mathrm{SNR} = 10^0$, our cutting-plane algorithm improved the three metrics faster than did the regularized estimation methods with the increasing number of data instances (Fig. 4d–f). When $\mathrm{SNR} = 10^1$, our cutting-plane algorithm



(a) Time vs. n

(b) Time vs. p

(c) Time vs.θ

(d) Time vs.γ

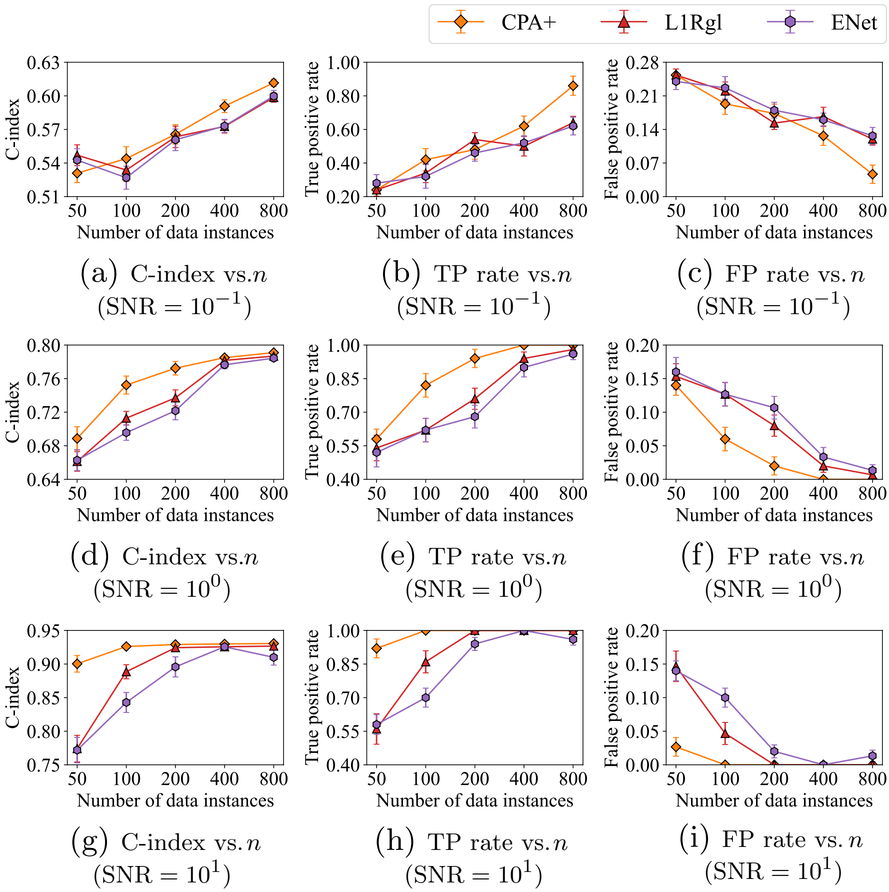
Fig. 3 Computation time with various parameter values (default: $n = 300$, $p = 20$, $\theta = 10$, $\gamma = 10^0$) for synthetic datasets (SNR $= 10^0$, $\theta^\star = 10$)

improved the three metrics even faster with the increasing number of data instances (Fig. 4g–i).

4.5 Results for real-world datasets

We used three real-world datasets provided by the Python `lifelines` library for survival analysis. Table 3 shows details of the datasets, where n and p are the numbers of data instances and candidate features, respectively. We omitted data instances with the same event time and removed variables unsuitable for prediction. Categorical variables were transformed into sets of dummy variables. Each dataset was randomly partitioned into training (70%) and testing (30%) datasets.

Table 4 shows the computational results for the real-world datasets, where the subset size and regularization parameters were tuned through hold-out validation in training datasets. Here, we selected $\theta \in \{4, 8, \dots, 40\}$ for the `canada` dataset, $\theta \in \{1, 2, \dots, 10\}$ for the `gbsg2` dataset, and $\theta \in \{2, 4, \dots, 20\}$ for the `lung` dataset. A computation of the cutting-plane algorithms was terminated if it did not complete within 100 s; in those cases, the best feasible solution found within 100 s was taken as the result. We repeated random dataset partition and performance evaluation ten times and calculated mean values. In the table, values in parentheses are standard errors for the C-index. The largest C-index values for each dataset are shown in bold.



(a) C-index vs.n (SNR $= 10^{-1}$)

(b) TP rate vs.n (SNR $= 10^{-1}$)

(c) FP rate vs.n (SNR $= 10^{-1}$)

(d) C-index vs.n (SNR $= 10^{0}$)

(e) TP rate vs.n (SNR $= 10^{0}$)

(f) FP rate vs.n (SNR $= 10^{0}$)

(g) C-index vs.n (SNR $= 10^{1}$)

(h) TP rate vs.n (SNR $= 10^{1}$)

(i) FP rate vs.n (SNR $= 10^{1}$)

Fig. 4 Performance metrics as a function of the number of data instances for synthetic datasets ($p = 20,\ \theta = \theta^\star = 5$)

Table 3 Real-world datasets

Name	n	p	Description
`canada`	844	44	History of Canadian senators in office
`gbsg2`	574	12	Observations from the GBSG2 study of 686 women
`lung`	149	23	Survival in patients with advanced lung cancer

Our cutting-plane algorithm with the quadratic approximation (CPA+) attained the best prediction accuracy (C-index) for the `gbsg2` and `lung` datasets. The regularized estimation methods (L1Rgl and ENet) had the best C-index value for the `canada` dataset, whereas our cutting-plane algorithm achieved comparable prediction accuracy with a much smaller number of features. These results show that our cutting-plane algorithm can work well on real-world datasets.

Table 4 Results for the real-world datasets

Dataset	n	p	Method	C-index	$\lvert\hat{S}\rvert$	#(Iter)	Time
canada	844	44	CPA	0.515 (± 0.005)	14.0	11864.7	4513.1
			CPA+	0.514 (± 0.006)	16.0	60541.5	4188.8
			L1Rgl	**0.517** (± 0.007)	39.5	0.0	6.4
			ENet	**0.517** (± 0.006)	40.6	0.0	54.2
gbsg2	574	12	CPA	0.680 (± 0.007)	6.6	1799.8	359.7
			CPA+	**0.684** (± 0.006)	6.3	1941.3	83.2
			L1Rgl	0.669 (± 0.006)	10.9	0.0	1.8
			ENet	0.675 (± 0.005)	11.2	0.0	15.9
lung	149	23	CPA	0.576 (± 0.015)	12.0	18540.3	1779.8
			CPA+	**0.588** (± 0.012)	11.6	36658.8	512.0
			L1Rgl	0.574 (± 0.011)	21.7	0.0	1.2
			ENet	0.565 (± 0.013)	21.8	0.0	10.3

5 Conclusion

We investigated estimation of sparse Cox PH models for survival analysis. For this purpose, we developed a cutting-plane algorithm that selects the best subset of features for the Cox PH model. To improve the computational efficiency of the cutting-plane algorithm, we applied quadratic approximation to the Fenchel conjugate function. We also effectively used the least-squares method to construct quadratic approximations that work well on each dataset.

In computational experiments conducted using synthetic and real-world datasets, our method was superior to the regularized estimation methods in terms of accuracy for both prediction and subset selection especially for low-dimensional datasets. Our quadratic approximation of the Fenchel conjugate function made the cutting-plane algorithm much faster and successfully maintained good out-of-sample prediction performance.

Our study broadens the potential of MIO methods for sparse estimation in survival analysis. Although our method is likely to find high-quality sparse solutions for the Cox PH model, applying it to large datasets is computationally expensive. It is thus more practical to choose between our method and heuristic algorithms according to the task at hand.

If we find lower and upper approximations of the univariate functions (37) and (38), we can provide a guarantee on the approximation accuracy of obtained solutions. It is probably difficult to make such quadratic approximations of high accuracy because the domains of the original univariate functions (37) and (38) are unbounded; however, this will be an interesting direction of future research.

A future direction of study will be to improve the algorithm's performance for sparse estimations. For example, we can use stochastic algorithms (Bertsimas et al. 2021; Kudo et al. 2020) to quickly find high-quality solutions to the upper- and lower-level problems. We can also impose appropriate constraints (Deng et al. 2018) on regression coefficients to enhance the generalization performance of constrained

Cox PH models. Another direction of future research will be to extend our method to other statistical models for survival analysis.

Appendix 1: Proof of Theorem 5

It is clear from Eq. (32) that

$$\alpha_i = \sum_{k=i}^{n} \alpha_k - \sum_{k=i+1}^{n} \alpha_k = \zeta_i - \zeta_{i+1} \quad (i \in [n]). \tag{46}$$

Therefore, it follows from Eqs. (32) and (46) that the domain constraints (5)–(7) on $\boldsymbol{\alpha} \in \mathbb{R}^n$ can be converted into Eqs. (34)–(36) on $\boldsymbol{\zeta} \in \mathbb{R}^{n+1}$.

It is clear from Eq. (46) that the first term of the Fenchel conjugate function (4) can be rewritten as

$$\sum_{i=1}^{n} (\delta_i + \alpha_i) \log(\delta_i + \alpha_i) = \sum_{i=1}^{n} (\delta_i + \zeta_i - \zeta_{i+1}) \log(\delta_i + \zeta_i - \zeta_{i+1}).$$

The remaining terms of the Fenchel conjugate function (4) are transformed as follows:

$$\begin{aligned}
&\sum_{i=1}^{n-1} \alpha_i \log\left(\frac{\prod_{k=i+1}^{n} \left(\delta_k + \sum_{\ell=k}^{n} \alpha_\ell\right)}{\prod_{k=i+1}^{n} \sum_{\ell=k}^{n} \alpha_\ell} \right) - \sum_{i=1}^{n} \delta_i \log\left(\delta_i + \sum_{k=i}^{n} \alpha_k \right) \\
&= \sum_{i=1}^{n-1} (\zeta_i - \zeta_{i+1}) \log\left(\frac{\prod_{k=i+1}^{n} \left(\delta_k + \zeta_k\right)}{\prod_{k=i+1}^{n} \zeta_k} \right) - \sum_{i=1}^{n} \delta_i \log\left(\delta_i + \zeta_i\right) \quad \because \text{Eqs. (32) and (46)} \\
&= \sum_{i=1}^{n-1} (\zeta_i - \zeta_{i+1}) \sum_{k=i+1}^{n} (\log(\delta_k + \zeta_k) - \log(\zeta_k)) - \sum_{i=1}^{n} \delta_i \log\left(\delta_i + \zeta_i\right).
\end{aligned} \tag{47}$$

The first term of Eq. (47) is further transformed as follows:

$$\begin{aligned}
&\sum_{i=1}^{n-1} (\zeta_i - \zeta_{i+1}) \sum_{k=i+1}^{n} \underbrace{(\log(\delta_k + \zeta_k) - \log(\zeta_k))}_{\tau_k} \\
&= \sum_{i=1}^{n-1} \zeta_i \sum_{k=i+1}^{n} \tau_k - \sum_{i=0}^{n-1} \zeta_{i+1} \sum_{k=i+1}^{n} \tau_k \quad \because \text{Eq. (34)} \\
&= \sum_{i=1}^{n-1} \zeta_i \sum_{k=i+1}^{n} \tau_k - \sum_{i=1}^{n-1} \zeta_i \sum_{k=i}^{n} \tau_k - \zeta_n \tau_n \\
&= -\sum_{i=1}^{n} \zeta_i \tau_i = -\sum_{i=1}^{n} \zeta_i (\log(\delta_i + \zeta_i) - \log(\zeta_i)).
\end{aligned}$$

Therefore, Eq. (47) is rewritten as

$$
\begin{aligned}
&-\sum_{i=1}^{n}\zeta_i(\log(\delta_i+\zeta_i)-\log(\zeta_i))-\sum_{i=1}^{n}\delta_i\log\left(\delta_i+\zeta_i\right)\\
&=\sum_{i=1}^{n}\zeta_i\log(\zeta_i)-\sum_{i=1}^{n}(\delta_i+\zeta_i)\log(\delta_i+\zeta_i),
\end{aligned}
$$

which completes the proof.

Funding This work was supported by JSPS KAKENHI Grant Number JP21K04526 and JST SPRING Grant Number JPMJSP2124.

Data availability The datasets generated during and/or analyzed during the current study are available from the corresponding author on reasonable request.

Declarations

Conflict of interest The authors declare they have no financial interests.

References

Aalen O (1978) Nonparametric inference for a family of counting processes. Ann Stat 6(4):701–726

Arthanari TS, Dodge Y (1981) Mathematical programming in statistics. Wiley, Hoboken

Berk L, Bertsimas D (2019) Certifiably optimal sparse principal component analysis. Math Program Comput 11(3):381–420

Bertsimas D, King A (2016) An algorithmic approach to linear regression. Oper Res 64(1):2–16

Bertsimas D, King A (2017) Logistic regression: from art to science. Stat Sci 32(3):367–384

Bertsimas D, Li ML (2020) Scalable holistic linear regression. Oper Res Lett 48(3):203–208

Bertsimas D, King A, Mazumder R (2016) Best subset selection via a modern optimization lens. Ann Stat 44(2):813–852

Bertsimas D, Pauphilet J, Van Parys B (2021) Sparse classification: a scalable discrete optimization perspective. Mach Learn 110(11):3177–3209

Boyd S, Vandenberghe L (2004) Convex optimization. Cambridge University Press, Cambridge

Bradburn MJ, Clark TG, Love SB, Altman DG (2003) Survival analysis part III: multivariate data analysis–choosing a model and assessing its adequacy and fit. Br J Cancer 89(4):605–611

Breslow N (1974) Covariance analysis of censored survival data. Biometrics 30(1):89–99

Buckley J, James I (1979) Linear regression with censored data. Biometrika 66(3):429–436

Clark TG, Bradburn MJ, Love SB, Altman DG (2003) Survival analysis part IV: further concepts and methods in survival analysis. Br J Cancer 89(5):781–786

Cox DR (1972) Regression models and life-tables. J R Stat Soc Ser B (Methodol) 34(2):187–202

Cox DR (1975) Partial likelihood. Biometrika 62(2):269–276

Cozad A, Sahinidis NV, Miller DC (2014) Learning surrogate models for simulation-based optimization. AIChE J 60(6):2211–2227

Cutler SJ, Ederer F (1958) Maximum utilization of the life table method in analyzing survival. J Chronic Dis 8(6):699–712

Davidson-Pilon C (2019) Lifelines: survival analysis in Python. J Open Source Softw 4(40):1317

Demyanyk Y, Hasan I (2010) Financial crises and bank failures: a review of prediction methods. Omega 38(5):315–324

Deng L, Ding J, Liu Y, Wei C (2018) Regression analysis for the proportional hazards model with parameter constraints under case-cohort design. Comput Stat Data Anal 117:194–206

Efron B (1977) The efficiency of Cox’s likelihood function for censored data. J Am Stat Assoc 72(359):557–565

Fan J, Li R (2002) Variable selection for Cox’s proportional hazards model and frailty model. Ann Stat 30(1):74–99

Goeman JJ (2010) L1 penalized estimation in the Cox proportional hazards model. Biometr J 52(1):70–84

Gui J, Li H (2005) Penalized Cox regression analysis in the high-dimensional and low-sample size settings, with applications to microarray gene expression data. Bioinformatics 21(13):3001–3008

Harrell FE Jr, Lee KL, Mark DB (1996) Multivariable prognostic models: Issues in developing models, evaluating assumptions and adequacy, and measuring and reducing errors. Stat Med 15(4):361–387

Hastie T, Tibshirani R, Tibshirani RJ (2020) Best subset, forward stepwise or Lasso? Analysis and recommendations based on extensive comparisons. Stat Sci 35(4):579–592

Kamiya S, Miyashiro R, Takano Y (2019). Feature subset selection for the multinomial logit model via mixed-integer optimization. In: The 22nd international conference on artificial intelligence and statistics, PMLR, pp 1254–1263

Kaplan EL, Meier P (1958) Nonparametric estimation from incomplete observations. J Am Stat Assoc 53(282):457–481

Katzman JL, Shaham U, Cloninger A, Bates J, Jiang T, Kluger Y (2018) DeepSurv: personalized treatment recommender system using a Cox proportional hazards deep neural network. BMC Med Res Methodol 18(1):1–12

Klein JP, Moeschberger ML (2003) Survival analysis: techniques for censored and truncated data. Springer, New York

Kobayashi K, Takano Y, Nakata K (2021) Bilevel cutting-plane algorithm for cardinality-constrained mean-CVaR portfolio optimization. J Glob Optim 81(2):493–528

Kobayashi K, Takano Y, Nakata K (2023) Cardinality-constrained distributionally robust portfolio optimization. Eur J Oper Res 309(3):1173–1182

Konno H, Yamamoto R (2009) Choosing the best set of variables in regression analysis using integer programming. J Glob Optim 44(2):273–282

Kudo K, Takano Y, Nomura R (2020) Stochastic discrete first-order algorithm for feature subset selection. IEICE Trans Inf Syst 103(7):1693–1702

Lane WR, Looney SW, Wansley JW (1986) An application of the Cox proportional hazards model to bank failure. J Bank Financ 10(4):511–531

Lee S, Lim H (2019) Review of statistical methods for survival analysis using genomic data. Genom Inform 17(4):e41

Li R, Chang C, Justesen JM, Tanigawa Y, Qian J, Hastie T, Tibshirani R (2022) Fast Lasso method for large-scale and ultrahigh-dimensional Cox model with applications to UK Biobank. Biostatistics 23(2):522–540

Maldonado S, Pérez J, Weber R, Labbé M (2014) Feature selection for support vector machines via mixed integer linear programming. Inf Sci 279:163–175

Mazumder R, Radchenko P, Dedieu A (2023) Subset selection with shrinkage: sparse linear modeling when the SNR is low. Oper Res 71(1):129–147

Miyashiro R, Takano Y (2015a) Subset selection by Mallows’ C_p: a mixed integer programming approach. Expert Syst Appl 42(1):325–331

Miyashiro R, Takano Y (2015b) Mixed integer second-order cone programming formulations for variable selection in linear regression. Eur J Oper Res 247(3):721–731

Naganuma M, Takano Y, Miyashiro R (2019) Feature subset selection for ordered logit model via tangent-plane-based approximation. IEICE Trans Inf Syst 102(5):1046–1053

Nelson W (1972) Theory and applications of hazard plotting for censored failure data. Technometrics 14(4):945–966

Park MY, Hastie T (2007) L1-regularization path algorithm for generalized linear models. J R Stat Soc Ser B (Stat Methodol) 69(4):659–677

Park YW, Klabjan D (2020) Subset selection for multiple linear regression via optimization. J Glob Optim 77(3):543–574

Rosset S, Neumann E, Eick U, Vatnik N (2003) Customer lifetime value models for decision support. Data Min Knowl Discov 7(3):321–339

Saikia R, Barman MP (2017) A review on accelerated failure time models. Int J Stat Syst 12(2):311–322

Saishu H, Kudo K, Takano Y (2021) Sparse Poisson regression via mixed-integer optimization. PLoS One 16(4):e0249916

Sato T, Takano Y, Miyashiro R, Yoshise A (2016) Feature subset selection for logistic regression via mixed integer optimization. Comput Optim Appl 64(3):865–880

Sato T, Takano Y, Miyashiro R (2017) Piecewise-linear approximation for feature subset selection in a sequential logit model. J Oper Res Soc Jpn 60(1):1–14

Simon N, Friedman J, Hastie T, Tibshirani R (2011) Regularization paths for Cox's proportional hazards model via coordinate descent. J Stat Softw 39(5):1–13

Takano Y, Miyashiro R (2020) Best subset selection via cross-validation criterion. TOP 28(2):475–488

Tamura R, Kobayashi K, Takano Y, Miyashiro R, Nakata K, Matsui T (2017) Best subset selection for eliminating multicollinearity. J Oper Res Soc Jpn 60(3):321–336

Tamura R, Kobayashi K, Takano Y, Miyashiro R, Nakata K, Matsui T (2019) Mixed integer quadratic optimization formulations for eliminating multicollinearity based on variance inflation factor. J Glob Optim 73(2):431–446

Tamura R, Takano Y, Miyashiro R (2022) Feature subset selection for kernel SVM classification via mixed-integer optimization. arXiv preprint arXiv:2205.14325

Tibshirani R (1997) The Lasso method for variable selection in the Cox model. Stat Med 16(4):385–395

Tobin J (1958) Estimation of relationships for limited dependent variables. Econometr J Econometr Soc 26(1):24–36

Uno H, Cai T, Pencina MJ, D'Agostino RB, Wei LJ (2011) On the C-statistics for evaluating overall adequacy of risk prediction procedures with censored survival data. Stat Med 30(10):1105–1117

Ustun B, Rudin C (2016) Supersparse linear integer models for optimized medical scoring systems. Mach Learn 102(3):349–391

Van De Vijver MJ, He YD, Van't Veer LJ, Dai H, Hart AAM, Voskuil DW, Bernards R (2002) A gene-expression signature as a predictor of survival in breast cancer. N Engl J Med 347(25):1999–2009

Van den Poel D, Larivière B (2004) Customer attrition analysis for financial services using proportional hazard models. Eur J Oper Res 157(1):196–217

Van Wieringen WN, Kun D, Hampel R, Boulesteix AL (2009) Survival prediction using gene expression data: a review and comparison. Comput Stat Data Anal 53(5):1590–1603

Verweij PJ, Van Houwelingen HC (1994) Penalized likelihood in Cox regression. Stat Med 13(23–24):2427–2436

Vinzamuri B, Reddy CK (2013) Cox regression with correlation based regularization for electronic health records. In: 2013 IEEE 13th international conference on data mining, IEEE, pp 757–766

Wächter A, Biegler LT (2006) On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. Math Programm 106(1):25–57

Wang P, Li Y, Reddy CK (2019) Machine learning for survival analysis: a survey. ACM Comput Surv (CSUR) 51(6):1–36

Watanabe A, Tamura R, Takano Y, Miyashiro R (2023) Branch-and-bound algorithm for optimal sparse canonical correlation analysis. Expert Syst Appl 217:119530

Wilson CM, Li K, Sun Q, Kuan PF, Wang X (2021) Fenchel duality of Cox partial likelihood with an application in survival kernel learning. Artif Intell Med 116:102077

Zhang HH, Lu W (2007) Adaptive Lasso for Cox's proportional hazards model. Biometrika 94(3):691–703

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.