Cutting-plane algorithm for sparse estimation of the Cox proportional-hazards model

Hiroki Saishu[1], Kota Kudo[1] and Yuichi Takano[2*]

[1]Graduate School of Science and Technology, University of Tsukuba, 1-1-1 Tennodai, Tsukuba-shi, 305-8573, Ibaraki, Japan.
[2]Faculty of Engineering, Information and Systems, University of Tsukuba, 1-1-1 Tennodai, Tsukuba-shi, 305-8573, Ibaraki, Japan.

*Corresponding author(s). E-mail(s): ytakano@sk.tsukuba.ac.jp;

Abstract

Survival analysis is a family of statistical methods for analyzing event occurrence times. In this paper, we address the mixed-integer optimization approach to sparse estimation of the Cox proportional-hazards model for survival analysis. Specifically, we propose a high-performance cutting-plane algorithm based on reformulation of bilevel optimization for sparse estimation. 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 the L1-regularized estimation method in terms of accuracy for both prediction and subset selection. Moreover, our quadratic approximation of the Fenchel conjugate function accelerates the cutting-plane algorithm and improves the generalization performance of sparse Cox proportional-hazards models.

Keywords: Cox model, sparse estimation, cutting-plane algorithm, mixed-integer optimization, Fenchel conjugate, survival analysis

1 Introduction

1.1 Background

Survival analysis [30] is a family of statistical methods for analyzing survival time, which is length of time between the start of an observation and the occurrence of an event of interest. Successful applications of survival analysis can be found in various real-world domains [66], such as gene expression analyses [60, 63], customer relationship management [46, 61], and credit risk evaluations [19, 35]. One of the main challenges inherent to such 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 [30, 36, 66]. These methods are categorized into three types of statistical models [66]: 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), and that its parameters are tuned using tobit regression [57], Buckley-James regression [12], or an accelerated failure time (AFT) model [48]. By contrast, nonparametric models estimate a survival function without such assumptions regarding distributions; these include the Kaplan-Meier estimator [28], the Nelson-Aalen estimator [1, 43], and life-table analysis [17]. 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 [29, 66].

1.2 Related work

We focus on sparse estimation of the Cox proportional-hazards (PH) model [14], which is 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 [10, 13]. In particular, various regularization methods have been applied to sparse estimation of the Cox PH model. These include lasso (L_1-regularization) [24, 37, 56], SCAD [22], adaptive lasso [69], correlation-based regularization [64], and elastic net [23, 44, 51]. However, these regularized estimation methods, which produce biased estimates due to the regularization term, are likely to yield low-quality solutions.

We address the mixed-integer optimization (MIO) approach to sparse estimation. First proposed for linear regression in the 1970s [2], this approach has recently gained increased attention due to advances in optimization algorithms and computer hardware [5, 16, 26, 33, 59]. 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 [40, 41, 45, 52]. MIO-based methods for sparse estimation have been extended to logistic regression [6, 50], ordinal regression [42, 49], count regression [47], support vector machine [38, 55], dimensionality reduction [3, 67], and elimination of multicollinearity [4, 7, 53, 54].

Bertsimas et al. [9] 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. [27] 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 sparse estimation of the Cox PH model. To our knowledge, we are the first to apply the cutting-plane algorithm [9, 27] to sparse estimation for survival analysis. Using the Fenchel conjugate [68] 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 sparse estimation of the Cox PH model. 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 the L_1-regularized estimation method in terms of accuracy for both prediction and subset selection. Moreover, our quadratic approximation of the Fenchel conjugate function accelerates the cutting-plane algorithm and enhances the 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 $[n] := \{1, 2, \ldots, n\}$. 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 sparse estimation of the Cox PH model.

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 event (or censored) 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 data instances are numbered in order of the event time as

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

The event indicator is defined as

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

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

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

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

The partial likelihood [15] 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 order of the observed event time. In Eq. (1), we assume that there are no ties between event times of uncensored instances, whereas some approximation methods [11, 21] 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 loss function

$$\begin{aligned} L(\boldsymbol{y}) :=& \sum_{i=1}^{n} \delta_i \left(\log\left(\sum_{k=i}^{n} \exp(y_k)\right) - y_i\right) \\ =& \sum_{i=1}^{n} \delta_i \left(\log\left(\sum_{k=i}^{n} \exp\left(\boldsymbol{w}^\top \boldsymbol{x}_k\right)\right) - \boldsymbol{w}^\top \boldsymbol{x}_i\right), \end{aligned} \tag{2}$$

which is known to be convex (see, e.g., Section 3.1.5 in Boyd and Vandenberghe [8]).

To improve generalization performance, we use the L_2-regularization term [39]

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

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 [8] 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. [68]) 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. [68], where the inequalities in Eqs. (6)–(7) are strict. From Theorem 5, however, the Fenchel conjugate function (33) is guaranteed to be bounded by Eqs. (6)–(7), because

$$\zeta_i \log(\zeta_i) = \left(\sum_{k=i}^{n}\alpha_k\right)\log\left(\sum_{k=i}^{n}\alpha_k\right) \to 0 \quad \left(\sum_{k=i}^{n}\alpha_k \searrow 0\right),$$

$$(\delta_i+\zeta_i-\zeta_{i+1})\log(\delta_i+\zeta_i-\zeta_{i+1}) = (\delta_i+\alpha_i)\log(\delta_i+\alpha_i) \to 0 \quad (\delta_i+\alpha_i \searrow 0),$$

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

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

$$\begin{aligned} \underset{\boldsymbol{\alpha}\in\mathbb{R}^n}{\text{maximize}} \quad & -L^*(\boldsymbol{\alpha}) - \frac{\gamma}{2}\boldsymbol{\alpha}^\top \boldsymbol{X}\boldsymbol{X}^\top\boldsymbol{\alpha} && (8)\\ \text{subject to} \quad & \sum_{i=1}^{n}\alpha_i = 0, && (9)\\ & \sum_{k=i+1}^{n}\alpha_k \geq 0 \quad (i\in[n-1]), && (10)\\ & \delta_i+\alpha_i \geq 0 \quad (i\in[n]). && (11) \end{aligned}$$

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 [8], strong duality holds for this problem. The Lagrange dual of problem (12) is formulated as

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

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

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 sparse estimation of the Cox PH model as

$$\begin{aligned} \underset{(\boldsymbol{w},\boldsymbol{z}) \in \mathbb{R}^p \times \{0,1\}^p}{\text{minimize}} \quad & L(\boldsymbol{X}\boldsymbol{w}) + \frac{1}{2\gamma} \|\boldsymbol{w}\|^2 && (13) \\ \text{subject to} \quad & z_j = 0 \;\Rightarrow\; w_j = 0 \quad (j \in [p]), && (14) \\ & \sum_{j=1}^p z_j = \theta, && (15) \end{aligned}$$

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 by using indicator constraints implemented in modern optimization software. The logical implication (14) can also be represented as

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

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. [9], we thus consider reformulation of 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

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

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

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)–(19) becomes

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

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

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

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

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

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

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

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

From Theorem 2, the associated dual formulation is represented as

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

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

Note here that

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

which completes the proof. □

Theorem 3 allows us to redefine $f(\boldsymbol{z})$ as the optimal objective value of problem (20)–(23) for real-valued vector $\boldsymbol{z} \in [0,1]^p$. In this case, following Bertsimas et al. [9], we can see that $f(\boldsymbol{z})$ is a convex function with a subgradient given by

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

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

3 Cutting-plane algorithm

This section describes our cutting-plane algorithm for sparse estimation of the Cox PH model. 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 [9] to sparse estimation of the Cox PH model. Our algorithm, which is based on a reformulation of bilevel optimization, aims to solve the upper-level problem (16)–(17).

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

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

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

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

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

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

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 $(\boldsymbol{z}^{(k)}, \xi^{(k)})$ to problem (29)–(30).

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

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

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

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

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

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

Algorithm 1 summarizes our cutting-plane algorithm. Note that the surrogate upper-level problem (29)–(30) is a mixed-integer linear optimization problem, which can be solved to optimality using optimization software. Following Kobayashi et al. [31, 32], we can prove the finite convergence of the algorithm.

Theorem 4 (Kobayashi et al. [31, 32]) Algorithm 1 terminates in a finite number of iterations and outputs an ε-optimal solution to problem (16)–(17).

Proof See Kobayashi et al. [31, 32]. □

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 $\varepsilon$-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).

Define $\boldsymbol{\zeta} := (\zeta_i)_{i \in [n+1]} \in \mathbb{R}^{n+1}$ 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 $\boldsymbol{\zeta} \in \mathbb{R}^{n+1}$ 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 A. □

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 $\delta \in \{0, 1\}$. We approximate these univariate convex functions 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

$$L^*(\boldsymbol{\alpha}) \approx \tilde{L}^*(\boldsymbol{\alpha}) := \sum_{i=1}^{n} \tilde{f}_1(\alpha_i \mid \delta_i) + \sum_{i=1}^{n} \tilde{f}_2(\zeta_i \mid \delta_i)$$

$$= \sum_{i=1}^{n} \tilde{f}_1(\alpha_i \mid \delta_i) + \sum_{i=1}^{n} \tilde{f}_2 \left(\sum_{k=i}^{n} \alpha_k \,\middle|\, \delta_i \right). \quad \because \text{Eq. (32)}$$

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)–(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)–(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 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 $\boldsymbol{\zeta}^\star$:

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

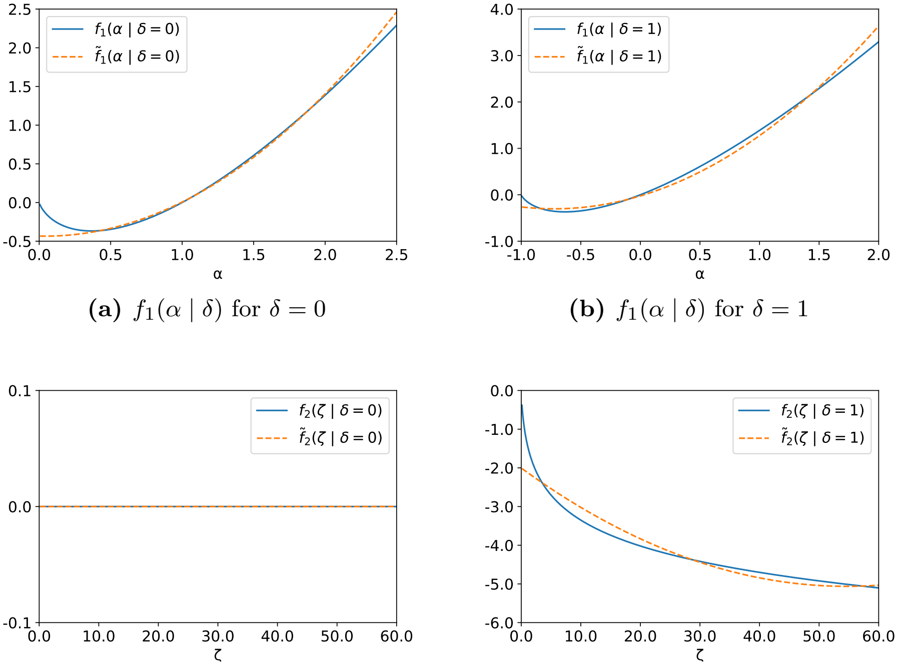
$$\underset{\boldsymbol{q}_2^{(\delta)} \in \mathbb{R}^3}{\text{minimize}} \quad \sum_{i \in \mathcal{N}_\delta} \left(f_2\left(\zeta_i^\star \mid \delta_i\right) - \left(q_{21}^{(\delta)} (\zeta_i^\star)^2 + q_{22}^{(\delta)} \zeta_i^\star + q_{23}^{(\delta)} \right) \right)^2, \tag{44}$$

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 [8].

Figure 1 shows examples of quadratic functions (dashed curves) for approximating the univariate convex functions (37)–(38) (solid curves). Here, quadratic approximations were computed using the least-squares method (43)–(44) for a synthetic dataset with $(\sigma^2, \rho) = (4.0, 0.70)$; see Section 4.3 for details of the dataset.

4 Computational results

This section evaluates the effectiveness of our method for sparse estimation through computational experiments using synthetic and real-world datasets.



(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 $f_2(\zeta \mid \delta)$

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 sparse estimation of the Cox PH model:

CPA: The cutting-plane algorithm (Algorithm 1);
CPA+: The cutting-plane algorithm (Algorithm 1) using the quadratic approximation problem (41)–(42), where quadratic functions (39)–(40) were determined by the least-squares method (43)–(44);
L1Rgl: L_1-regularized estimation [56].

We implemented these methods in Python. In the cutting-plane algorithms, the mixed-integer linear optimization problem (29)–(30) and convex quadratic optimization problem (41)–(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] [65]. 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 $\gamma \in \{10^0, 10^{-1}, 10^{-2}\}$ as the L_2-regularization parameter. We implemented regularized estimation methods using the Python `lifelines` library[3] [18], and tuned the L_1-regularization parameter such that the number of features with nonzero regression coefficients was θ.

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 to obtain a vector $\hat{\boldsymbol{w}} := (\hat{w}_j)_{j\in[p]} \in \mathbb{R}^p$ of regression coefficients. Here, regularization parameters 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 recall [62], defined as

$$\textbf{Recall} := \frac{|\mathcal{S}^\star \cap \hat{\mathcal{S}}|}{|\mathcal{S}^\star|},$$

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) [25, 58], a rank-correlation measure often used for censored survival data. The C-index is defined as

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

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.

4.3 Generation of synthetic datasets

Following previous studies [5, 26], we prepared synthetic datasets according to the following steps, with $p = 30$ set as the number of candidate features. First, we defined a vector of true coefficients and the index set of relevant features as

$$\boldsymbol{w}^\star := (1,0,0,0,0,0,1,0,0,0,0,0,\ldots,1,0,0,0,0,0)^\top \in \mathbb{R}^{30},$$

[1] https://www.gurobi.com/products/gurobi-optimizer/
[2] https://github.com/coin-or/Ipopt
[3] https://lifelines.readthedocs.io/en/latest/

$$\mathcal{S}^\star := \{1, 7, 13, 19, 25\}. \tag{45}$$

Next, we sampled feature vectors from a multivariate normal distribution as $\boldsymbol{x}_i \sim \mathrm{N}(\boldsymbol{0}, \boldsymbol{\Sigma})$, where $\boldsymbol{\Sigma} \in \mathbb{R}^{30\times 30}$ is the covariance matrix. The (i, j)th entry of $\boldsymbol{\Sigma}$ is $\rho^{|i-j|}$, where $\rho \in \{0.35, 0.70\}$ represents the correlation strength between features. We also sampled the error term from a univariate normal distribution as $\varepsilon_i \sim \mathrm{N}(0, \sigma^2)$, where $\sigma^2 \in \{1.0, 4.0, 9.0\}$ is the variance. We then calculated the hazard function with baseline $h_0(t) = 1$ as

$$h(\boldsymbol{x}_i, \varepsilon_i) := \exp\left(\frac{(\boldsymbol{w}^\star)^\top \boldsymbol{x}_i}{\sqrt{(\boldsymbol{w}^\star)^\top \boldsymbol{\Sigma} \boldsymbol{w}^\star}} + \varepsilon_i\right) \quad (i \in [n]).$$

Finally, we renumbered the data instances as

$$h(\boldsymbol{x}_1, \varepsilon_1) \geq h(\boldsymbol{x}_2, \varepsilon_2) \geq \cdots \geq h(\boldsymbol{x}_n, \varepsilon_n),$$

and specified event times and indicators as

$$(t_i, \delta_i) = \begin{cases} (i, 1) & \text{if } i \in [d], \\ (d+1, 0) & \text{otherwise} \end{cases} \quad (i \in [n]),$$

where $d \in [n]$ is the number of uncensored instances.

We set $n = 200$ as the number of training data instances and generated sufficiently many data instances for the testing datasets. We set $d := 0.7 \cdot n$ as the number of uncensored instances.

4.4 Results for synthetic datasets

Table 1 gives the computational results with the subset size parameter $\theta = 5$ for the synthetic datasets. We repeated data generation and performance evaluation 10 times and calculated mean values. In the table, values in parentheses are standard errors for the C-index and Recall. The largest C-index and Recall values for each problem instance are shown in bold.

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 (Recall). The cutting-plane algorithm without the quadratic approximation (CPA) required much longer computation times than CPA+ did. When $(\sigma^2, \rho) = (4.0, 0.35)$ for instance, the computation times spent by CPA and CPA+ were 151.0 s and 8.7 s, respectively. Although the L_1-regularized estimation method (L1Rgl) was very fast, its prediction and subset selection performance was often worse than with the cutting-plane algorithms.

These results show that our cutting-plane algorithm performs well in terms of accuracy for both prediction and subset selection. Moreover, our quadratic approximation of the Fenchel conjugate function not only accelerates the

Table 1: Result for the synthetic datasets ($n = 200, p = 30, \theta = 5$)

σ^2	ρ	Method	C-index	Recall	#Iter	Time
1.0	0.35	CPA	**0.743** (±0.003)	**0.980** (±0.019)	163.3	29.7
		CPA+	**0.743** (±0.003)	**0.980** (±0.019)	206.1	4.3
		L1Rgl	**0.743** (±0.003)	**0.980** (±0.019)	—	<0.1
	0.70	CPA	0.750 (±0.005)	**0.880** (±0.042)	338.1	59.3
		CPA+	**0.752** (±0.004)	**0.880** (±0.042)	514.1	12.0
		L1Rgl	0.745 (±0.005)	0.840 (±0.038)	—	<0.1
4.0	0.35	CPA	0.601 (±0.007)	0.640 (±0.062)	833.8	151.0
		CPA+	**0.612** (±0.008)	**0.760** (±0.068)	409.4	8.7
		L1Rgl	0.609 (±0.006)	0.720 (±0.058)	—	<0.1
	0.70	CPA	**0.613** (±0.004)	**0.440** (±0.074)	461.9	80.3
		CPA+	**0.613** (±0.005)	0.420 (±0.072)	171.1	3.7
		L1Rgl	0.604 (±0.004)	0.420 (±0.060)	—	<0.1
9.0	0.35	CPA	0.550 (±0.005)	0.440 (±0.055)	852.9	155.1
		CPA+	**0.556** (±0.005)	**0.500** (±0.071)	919.1	19.9
		L1Rgl	0.547 (±0.004)	0.440 (±0.055)	—	<0.1
	0.70	CPA	0.567 (±0.004)	0.300 (±0.051)	185.6	31.9
		CPA+	**0.569** (±0.004)	**0.320** (±0.058)	156.6	3.7
		L1Rgl	0.556 (±0.005)	0.260 (±0.049)	—	<0.1

cutting-plane algorithm but also aids in achieving high generalization performance; this observation is consistent with computational results reported by Kamiya et al. [27] for the multinomial logit model.

4.5 Results for real-world datasets

We used three real-world datasets provided by the Python `lifelines` library for survival analysis. Table 2 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. The first 300 instances (in the `canada` and `gbsg2` datasets) and 30 variables (in the `canada` dataset) were extracted, and each dataset was randomly partitioned into training (70%) and testing (30%) datasets.

Table 3 shows the computational results with the subset size parameter $\theta = 5$ for the real-world datasets. We repeated random dataset partition and performance evaluations 10 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.

Our cutting-plane algorithm with quadratic approximation (CPA+) attained the best prediction accuracy (C-index) for the `canada` and `gbsg2` datasets, whereas the cutting-plane algorithm without quadratic approximation (CPA) had the best C-index value for the `lung` dataset. By contrast, the

Table 2: Real-world datasets

Name	n	p	Description
canada	300	30	History of Canadian senators in office
gbsg2	300	12	Observations from the GBSG2 study of 686 women
lung	149	23	Survival in patients with advanced lung cancer

Table 3: Results for the real-world datasets ($\theta = 5$)

Dataset	n	p	Method	C-index	#Iter	Time
canada	300	30	CPA	0.514 (±0.011)	30.9	5.4
			CPA+	**0.526** (±0.019)	84.2	3.7
			L1Rgl	0.508 (±0.006)	—	<0.1
gbsb2	300	12	CPA	0.690 (±0.023)	10.0	1.6
			CPA+	**0.693** (±0.022)	18.6	0.4
			L1Rgl	0.633 (±0.022)	—	<0.1
lung	149	23	CPA	**0.582** (±0.045)	82.3	9.5
			CPA+	0.568 (±0.043)	222.3	3.2
			L1Rgl	0.571 (±0.039)	—	<0.1

L_1-regularized estimation method (L1Rgl) provided poor prediction accuracy for all these datasets. These results show that our cutting-plane algorithm can work well on real-world datasets.

5 Conclusion

We investigated sparse estimation of the Cox PH model 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 L_1-regularized estimation in terms of accuracy for both prediction and subset selection. Our quadratic approximation of the Fenchel conjugate function made the cutting-plane algorithm much faster and successfully improved the 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.

A future direction of study will be to improve the algorithm's performance for sparse estimations. For example, we can use stochastic algorithms [9, 34] to quickly find approximate solutions to the upper- and lower-level problems. We can also impose appropriate constraints [20] 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 A Proof of Theorem 5

It is clear from Eq. (32) that

$$\alpha_i = \sum_{k=i}^{n} \alpha_k - \sum_{k=i+1}^{n} \alpha_k = \zeta_i - \zeta_{i+1} \quad (i \in [n]). \tag{A1}$$

Therefore, it follows from Eqs. (32) and (A1) 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. (A1) 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:

$$\begin{aligned}
&\sum_{i=1}^{n-1} \alpha_i \log\left(\frac{\prod_{k=i+1}^{n}\left(\delta_k + \sum_{\ell=k}^{n} \alpha_\ell\right)}{\prod_{k=i+1}^{n} \sum_{\ell=k}^{n} \alpha_\ell}\right) - \sum_{i=1}^{n} \delta_i \log\left(\delta_i + \sum_{k=i}^{n} \alpha_k\right) \\
=& \sum_{i=1}^{n-1} (\zeta_i - \zeta_{i+1}) \log\left(\frac{\prod_{k=i+1}^{n}(\delta_k + \zeta_k)}{\prod_{k=i+1}^{n} \zeta_k}\right) - \sum_{i=1}^{n} \delta_i \log(\delta_i + \zeta_i) \qquad \because \text{Eqs. (32) and (A1)} \\
=& \sum_{i=1}^{n-1} (\zeta_i - \zeta_{i+1}) \sum_{k=i+1}^{n} \left(\log(\delta_k + \zeta_k) - \log(\zeta_k)\right) - \sum_{i=1}^{n} \delta_i \log(\delta_i + \zeta_i). \qquad \text{(A2)}
\end{aligned}$$

The first term of Eq. (A2) is further transformed as follows:

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

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

Therefore, Eq. (A2) 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.

References

[1] Aalen, O. (1978). Nonparametric inference for a family of counting processes. The Annals of Statistics, 6(4), 701–726.

[2] Arthanari, T. S., & Dodge, Y. (1981). Mathematical Programming in Statistics, Wiley.

[3] Berk, L., & Bertsimas, D. (2019). Certifiably optimal sparse principal component analysis. Mathematical Programming Computation, 11(3), 381–420.

[4] Bertsimas, D., & King, A. (2016). An algorithmic approach to linear regression. Operations Research, 64(1), 2–16.

[5] Bertsimas, D., King, A., & Mazumder, R. (2016). Best subset selection via a modern optimization lens. The annals of statistics, 44(2), 813–852.

[6] Bertsimas, D., & King, A. (2017). Logistic regression: From art to science. Statistical Science, 32(3), 367–384.

[7] Bertsimas, D., & Li, M. L. (2020). Scalable holistic linear regression. Operations Research Letters, 48(3), 203–208.

[8] Boyd, S., & Vandenberghe, L. (2004). Convex optimization. Cambridge university press.

[9] Bertsimas, D., Pauphilet, J., & Van Parys, B. (2021). Sparse classification: a scalable discrete optimization perspective. Machine Learning, 110(11), 3177–3209.

[10] Bradburn, M. J., Clark, T. G., Love, S. B., & Altman, D. G. (2003). Survival analysis Part III: multivariate data analysis—choosing a model

and assessing its adequacy and fit. British journal of cancer, 89(4), 605–611.

[11] Breslow, N. (1974). Covariance analysis of censored survival data. Biometrics, 30(1), 89–99.

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

[13] Clark, T. G., Bradburn, M. J., Love, S. B., & Altman, D. G. (2003). Survival analysis part IV: further concepts and methods in survival analysis. British journal of cancer, 89(5), 781–786.

[14] Cox, D. R. (1972). Regression models and life]tables. Journal of the Royal Statistical Society: Series B (Methodological), 34(2), 187–202.

[15] Cox, D. R. (1975). Partial likelihood. Biometrika, 62(2), 269–276.

[16] Cozad, A., Sahinidis, N. V., & Miller, D. C. (2014). Learning surrogate models for simulation]based optimization. AIChE Journal, 60(6), 2211–2227.

[17] Cutler, S. J., & Ederer, F. (1958). Maximum utilization of the life table method in analyzing survival. Journal of chronic diseases, 8(6), 699–712.

[18] Davidson-Pilon, C. (2019). lifelines: survival analysis in Python. Journal of Open Source Software, 4(40), 1317.

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

[20] Deng, L., Ding, J., Liu, Y., & Wei, C. (2018). Regression analysis for the proportional hazards model with parameter constraints under case-cohort design. Computational Statistics & Data Analysis, 117, 194–206.

[21] Efron, B. (1977). The efficiency of Cox's likelihood function for censored data. Journal of the American statistical Association, 72(359), 557–565.

[22] Fan, J., & Li, R. (2002). Variable selection for Cox's proportional hazards model and frailty model. The Annals of Statistics, 30(1), 74–99.

[23] Goeman, J. J. (2010). L1 penalized estimation in the Cox proportional hazards model. Biometrical journal, 52(1), 70–84.

[24] 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.

[25] Harrell Jr, F. E., Lee, K. L., & Mark, D. B. (1996). Multivariable prognostic models: issues in developing models, evaluating assumptions and adequacy, and measuring and reducing errors. Statistics in Medicine, 15(4), 361–387.

[26] Hastie, T., Tibshirani, R., & Tibshirani, R. J. (2020). Best subset, forward stepwise or lasso? Analysis and recommendations based on extensive comparisons. Statistical Science, 35(4), 579–592.

[27] Kamiya, S., Miyashiro, R., & Takano, Y. (2019, April). Feature subset selection for the multinomial logit model via mixed-integer optimization. In The 22nd International Conference on Artificial Intelligence and Statistics (pp. 1254–1263). PMLR.

[28] Kaplan, E. L., & Meier, P. (1958). Nonparametric estimation from incomplete observations. Journal of the American statistical association, 53(282), 457–481.

[29] Katzman, J. L., 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 medical research methodology, 18(1), 1–12.

[30] Klein, J. P., & Moeschberger, M. L. (2003). Survival analysis: techniques for censored and truncated data. New York: Springer.

[31] Kobayashi, K., Takano, Y., & Nakata, K. (2021). Bilevel cutting-plane algorithm for cardinality-constrained mean-CVaR portfolio optimization. Journal of Global Optimization, 81(2), 493–528.

[32] Kobayashi, K., Takano, Y., & Nakata, K. (2021). Cardinality-constrained Distributionally Robust Portfolio Optimization. arXiv preprint arXiv:2112.12454.

[33] Konno, H., & Yamamoto, R. (2009). Choosing the best set of variables in regression analysis using integer programming. Journal of Global Optimization, 44(2), 273–282.

[34] Kudo, K., Takano, Y., & Nomura, R. (2020). Stochastic discrete first-order algorithm for feature subset selection. IEICE Transactions on Information and Systems, 103(7), 1693–1702.

[35] Lane, W. R., Looney, S. W., & Wansley, J. W. (1986). An application of the Cox proportional hazards model to bank failure. Journal of Banking & Finance, 10(4), 511–531.

[36] Lee, S., & Lim, H. (2019). Review of statistical methods for survival analysis using genomic data. Genomics & informatics, 17(4), e41.

[37] Li, R., Chang, C., Justesen, J. M., 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.

[38] Maldonado, S., Pérez, J., Weber, R., & Labbé, M. (2014). Feature selection for support vector machines via mixed integer linear programming. Information Sciences, 279, 163–175.

[39] Mazumder, R., Radchenko, P., & Dedieu, A. (2017). Subset selection with shrinkage: Sparse linear modeling when the SNR is low. arXiv preprint arXiv:1708.03288.

[40] Miyashiro, R., & Takano, Y. (2015). Subset selection by Mallows' C_p: A mixed integer programming approach. Expert Systems with Applications, 42(1), 325–331.

[41] Miyashiro, R., & Takano, Y. (2015). Mixed integer second-order cone programming formulations for variable selection in linear regression. European Journal of Operational Research, 247(3), 721–731.

[42] Naganuma, M., Takano, Y., & Miyashiro, R. (2019). Feature subset selection for ordered logit model via tangent-plane-based approximation. IEICE Transactions on Information and Systems, 102(5), 1046–1053.

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

[44] Park, M. Y., & Hastie, T. (2007). L1-regularization path algorithm for generalized linear models. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 69(4), 659–677.

[45] Park, Y. W., & Klabjan, D. (2020). Subset selection for multiple linear regression via optimization. Journal of Global Optimization, 77(3), 543–574.

[46] Rosset, S., Neumann, E., Eick, U., & Vatnik, N. (2003). Customer lifetime value models for decision support. Data mining and knowledge discovery, 7(3), 321–339.

[47] Saishu, H., Kudo, K., & Takano, Y. (2021). Sparse Poisson regression via mixed-integer optimization. Plos one, 16(4), e0249916.

[48] Saikia, R., & Barman, M. P. (2017). A review on accelerated failure time models. Int J Stat Syst, 12(2), 311–322.

[49] Sato, T., Takano, Y., & Miyashiro, R. (2017). Piecewise-linear approximation for feature subset selection in a sequential logit model. Journal of the Operations Research Society of Japan, 60(1), 1–14.

[50] Sato, T., Takano, Y., Miyashiro, R., & Yoshise, A. (2016). Feature subset selection for logistic regression via mixed integer optimization. Computational Optimization and Applications, 64(3), 865–880.

[51] Simon, N., Friedman, J., Hastie, T., & Tibshirani, R. (2011). Regularization paths for Cox's proportional hazards model via coordinate descent. Journal of statistical software, 39(5), 1–13.

[52] Takano, Y., & Miyashiro, R. (2020). Best subset selection via cross-validation criterion. Top, 28(2), 475–488.

[53] Tamura, R., Kobayashi, K., Takano, Y., Miyashiro, R., Nakata, K., & Matsui, T. (2017). Best subset selection for eliminating multicollinearity. Journal of the Operations Research Society of Japan, 60(3), 321–336.

[54] 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. Journal of Global Optimization, 73(2), 431–446.

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

[56] Tibshirani, R. (1997). The lasso method for variable selection in the Cox model. Statistics in medicine, 16(4), 385–395.

[57] Tobin, J. (1958). Estimation of relationships for limited dependent variables. Econometrica: journal of the Econometric Society, 26(1), 24–36.

[58] Uno, H., Cai, T., Pencina, M. J., D'Agostino, R. B., & Wei, L. J. (2011). On the C-statistics for evaluating overall adequacy of risk prediction procedures with censored survival data. Statistics in Medicine, 30(10), 1105–1117.

[59] Ustun, B., & Rudin, C. (2016). Supersparse linear integer models for optimized medical scoring systems. Machine Learning, 102(3), 349–391.

[60] Van De Vijver, M. J., He, Y. D., Van't Veer, L. J., Dai, H., Hart, A. A. M., Voskuil, D. W., ... & Bernards, R. (2002). A gene-expression signature

as a predictor of survival in breast cancer. The New England Journal of Medicine, 347(25), 1999–2009.

[61] Van den Poel, D., & Larivière, B. (2004). Customer attrition analysis for financial services using proportional hazard models. European journal of operational research, 157(1), 196–217.

[62] Van Rijsbergen, C. J. (1979). Information Retrieval, 2nd edition. Butterworth–Heinemann.

[63] Van Wieringen, W. N., Kun, D., Hampel, R., & Boulesteix, A. L. (2009). Survival prediction using gene expression data: a review and comparison. Computational statistics & data analysis, 53(5), 1590–1603.

[64] Vinzamuri, B., & Reddy, C. K. (2013, December). Cox regression with correlation based regularization for electronic health records. In 2013 IEEE 13th International Conference on Data Mining (pp. 757–766). IEEE.

[65] Wächter, A., & Biegler, L. T. (2006). On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. Mathematical Programming, 106(1), 25–57.

[66] Wang, P., Li, Y., & Reddy, C. K. (2019). Machine learning for survival analysis: A survey. ACM Computing Surveys (CSUR), 51(6), 1–36.

[67] Watanabe, A., Tamura, R., Takano, Y., & Miyashiro, R. (2021). Branch-and-bound Algorithm for Optimal Sparse Canonical Correlation Analysis. Optimization Online.

[68] Wilson, C. M., Li, K., Sun, Q., Kuan, P. F., & Wang, X. (2021). Fenchel duality of Cox partial likelihood with an application in survival kernel learning. Artificial Intelligence in Medicine, 116, 102077.

[69] Zhang, H. H., & Lu, W. (2007). Adaptive Lasso for Cox's proportional hazards model. Biometrika, 94(3), 691–703.