Another Look at Linear Programming for Feature Selection via Methods of Regularization [1]

Yonggang Yao, *SAS Institute Inc.*
Yoonkyung Lee, *The Ohio State University*

Technical Report No. 800r

April, 2010

Department of Statistics
The Ohio State University
1958 Neil Avenue
Columbus, OH 43210-1247

[1]This is the second revision of Technical Report 800, Department of Statistics, The Ohio State University.

Another Look at Linear Programming for Feature Selection via Methods of Regularization

Yonggang Yao
SAS Institute Inc.
100 SAS Campus Dr, Cary, NC 27513, USA
AND
Yoonkyung Lee
Department of Statistics
The Ohio State University
1958 Neil Ave, Columbus, OH 43210, USA

Abstract

We consider statistical procedures for feature selection defined by a family of regularization problems with convex piecewise linear loss functions and penalties of l_1 nature. Many known statistical procedures (e.g. quantile regression and support vector machines with l_1 norm penalty) are subsumed under this category. Computationally, the regularization problems are linear programming (LP) problems indexed by a single parameter, which are known as 'parametric cost LP' or 'parametric right-hand-side LP' in the optimization theory. Exploiting the connection with the LP theory, we lay out general algorithms, namely, the simplex algorithm and its variant for generating regularized solution paths for the feature selection problems. The significance of such algorithms is that they allow a complete exploration of the model space along the paths and provide a broad view of persistent features in the data. The implications of the general path-finding algorithms are outlined for a few statistical procedures, and they are illustrated with numerical examples.

Keywords: Grouped Regularization, l_1-Norm Penalty, Parametric Linear Programming, Quantile Regression, Simplex Method, Structured Learning, Support Vector Machines

1 Introduction

Regularization methods cover a wide range of statistical procedures for estimation and prediction, and they have been used in many modern applications. To name a few, examples are ridge regression (Hoerl and Kennard, 1970), the LASSO regression (Tibshirani, 1996), smoothing splines (Wahba, 1990), and support vector machines (SVM) (Vapnik, 1998).

Given a training data set, $\{(y_i, \mathbf{x}_i) : \mathbf{x}_i \in \mathcal{R}^p; i = 1, \cdots, n\}$, many statistical problems can be phrased as the problem of finding a functional relationship between the covariates, $\mathbf{x} \in \mathcal{R}^p$, and the response y based on the observed pairs. For example, a regularization method for prediction looks for a model or a rule $f(\mathbf{x}; \boldsymbol{\beta})$ with unknown parameters $\boldsymbol{\beta}$ that minimizes prediction error over the training data while controlling its model complexity. To be precise, let $\mathcal{L}(y, f(\mathbf{x}; \boldsymbol{\beta}))$ be a convex loss function for the prediction error of f over a case $(y, \mathbf{x})$ and $J(f)$ be a convex penalty functional that measures the model complexity of

f. Formally, the solution to a regularization problem is defined to be f with the model parameters $\hat{\boldsymbol{\beta}}$ that minimizes

$$\sum_{i=1}^{n} \mathcal{L}(y_i, f(\mathbf{x}_i; \boldsymbol{\beta})) + \lambda \cdot J(f), \tag{1}$$

where $\lambda \geq 0$ is a pre-specified regularization parameter. The λ determines the trade-off between the prediction error and the model complexity, and thus the quality of the solution highly depends on the choice of λ. Identification of a proper value of the regularization parameter for model selection or a proper range for model averaging is a critical statistical problem. Note that $\hat{\boldsymbol{\beta}}(\lambda)$ is a function of λ. As in (1), each regularization method defines a continuum of optimization problems indexed by a tuning parameter. In most cases, the solution as a function of the tuning parameter is expected to change continuously with λ. This allows for the possibility of complete exploration of the model space as λ varies, and computational savings if (1) is to be optimized for multiple values of λ.

Alternatively, the regularization problem in (1) can be formulated to bound the model complexity. In this complexity-bounded formulation, the optimal parameters are sought by minimizing:

$$\sum_{i=1}^{n} \mathcal{L}(y_i, f(\mathbf{x}_i; \boldsymbol{\beta})) \text{ subject to } J(f) \leq s, \tag{2}$$

where s is an upper bound of the complexity.

For a certain combination of the loss $\mathcal{L}$ and the complexity measure J, it is feasible to generate the entire solution path of the regularization problem. Here, the path refers to the entire set of solutions to the regularization problem, for instance, $\hat{\boldsymbol{\beta}}(\lambda)$ in (1) as a function of λ (or $\hat{\boldsymbol{\beta}}(s)$ in (2) as a function of s). Some pairs of the loss and the complexity are known to allow such fast and efficient path finding algorithms; for instance, LARS (Efron et al., 2004), the standard binary SVM (Hastie et al., 2004), the multi-category SVM (Lee and Cui, 2006), and the l_1-norm quantile regression (Li and Zhu, 2008). Rosset and Zhu (2007) study general conditions for the combination of $\mathcal{L}$ and J such that solutions indexed by a regularization parameter are piecewise linear and thus can be sequentially characterized. They provide generic path-finding algorithms under some appropriate assumptions on $\mathcal{L}$ and J.

In this paper, we focus on an array of regularization methods aimed for feature selection with penalties of l_1 nature and piecewise linear loss functions. Many existing procedures are subsumed under this category. Examples include the l_1-norm SVM (Zhu et al., 2004) and its extension to the multi-class case (Wang and Shen, 2006), l_1-norm quantile regression (Li and Zhu, 2008), Sup-norm multi-category SVM (Zhang et al., 2006), the functional component selection step (called "θ-step") for structured multi-category SVM (Lee et al., 2006), and the Dantzig selector (Candes and Tao, 2007). We also note that the ϵ-insensitive loss in the SVM regression (Vapnik, 1998) fits into the category of a piecewise linear loss. As for the penalty, the sup norm gives rise to a linear penalty just as the l_1 norm in general, and so does a certain combination of the l_1 norm and the sup norm for desired grouping and clustering of features such as OSCAR penalty (Bondell and Reich, 2008).

There is a great commonality among these methods. That is, computationally the associated optimization problems are all linear programming (LP) problems indexed by a single regularization parameter. This family of LP problems are known as the *parametric cost* linear programming and have long been studied in the optimization theory. There already exist general algorithms for generating the solution paths of parametric LP in the literature. See Saaty and Gass (1954) and Gass and Saaty (1955a,b) for example. Despite the commonality, so far, only case-by-case treatments of computation for some of the procedures are available as

in Zhu et al. (2004); Li and Zhu (2008) and Wang and Shen (2006). Although Wang and Shen (2006) notice that those solution path algorithms have fundamental connections with the *parametric right-hand-side* LP (see (8) for the definition), such connections have not been adequately explored for other problems with full generality. As noted, Rosset and Zhu (2007) have a comprehensive take on the computational properties of regularized solutions. However, they did not tap into the LP theory for general treatments of the problems of the current focus. Rather, their approach centers on methods with loss functions of certain forms which need notion of residual and the l_1 norm penalty primarily, and adheres to a specific structure of the associated computational problems. With the LP formalism, the scope of related methods and computational problems to handle can be broader, and their treatment can be far more general.

The goal of this paper is to make it more explicit the link between the parametric LP and a family of computational problems arising in statistics for feature selection via regularization and put those feature selection problems in perspectives. Linear programming techniques, in fact, have been used in statistics for many other applications as well. For example, the least absolute deviation (LAD) regression, also known as L_1 regression in robust statistics, involves LP. See Wagner (1959); Fisher (1961); Bloomfield and Steiger (1980) for reference and also Bloomfield and Steiger (1983) for historical background, algorithms, and comprehensive literature on the subject. More generally, quantile regression entails LP and parametric LP, in particular, when regression fits for every quantile parameter are sought. See Chapter 6 of Koenker (2005) and references therein for computational aspects of quantile regression. The main focus of this paper is on parametric LP as a computational device to systematically explore a potentially large model space with a modular treatment of each feature selection method under consideration. To this end, we pull together relevant results from the linear programming literature and summarize them in an accessible and self-contained fashion.

Section 2 begins with an overview of the standard LP and parametric LP problems, and gives a brief account of the optimality conditions for their solutions. Section 3 presents the simplex algorithm and the tableau-simplex algorithm for finding the entire solution paths of the parametric LP problems. Section 4 describes a few examples of LP for feature selection, paraphrasing their computational elements in the LP terms. A detailed comparison of the simplex algorithm with the existing algorithm for the l_1-norm SVM (Zhu et al., 2004) is given in Section 5, highlighting the generality of the proposed approach. Numerical examples and data application of the algorithm follow in Section 6 for illustration. Technical proofs except for the key theorems are collected into Appendix.

2 Linear Programming

Linear programming (LP) is one of the cornerstones of the optimization theory. Since the publication of the simplex algorithm by Dantzig in 1947, there has been a wide range of LP applications in operation research, microeconomics, business management, and many other engineering fields. We give an overview of LP here and describe the optimality conditions of the LP solution pertinent to our discussion of path-finding algorithms later. Some properties of the LP to be described are well known in the optimization literature, but they are included here for completeness along with their proofs. Our treatment of LP closely follows that in standard references. See Dantzig (1951); Murty (1983); Gill et al. (1991); Vanderbei (1997), and Bertsimas and Tsitsiklis (1997). Some LP references contain discussions of the parametric LP; see Murty (1983) and Bertsimas and Tsitsiklis (1997), for example, and Gass and Saaty (1955a,b); Ǵal (1979) for earlier references. The readers are referred to them and references therein for more complete discussions.

Section 2.1 reviews basic notions in LP to mathematically characterize the optimality of a solution, directly based on Section 3.1 *Optimality Conditions* of Bertsimas and Tsitsiklis (1997). Section 2.2 describes

the important implications of the LP optimality condition for the parametric LP, mainly from Murty (1983), Section 8.2 *The parametric cost simplex algorithm*.

2.1 Standard Linear Programs

A standard form of LP is

$$\begin{cases} \min\limits_{\boldsymbol{z} \in \mathcal{R}^N} & \mathbf{c}'\boldsymbol{z} \\ \text{s.t.} & \mathbf{A}\boldsymbol{z} = \mathbf{b} \\ & \boldsymbol{z} \geq \mathbf{0}, \end{cases} \tag{3}$$

where $\boldsymbol{z}$ is an N-vector of variables, $\mathbf{c}$ is a fixed N-vector, $\mathbf{b}$ is a fixed M-vector, and $\mathbf{A}$ is an $M \times N$ fixed matrix. Without loss of generality, it is assumed that $M \leq N$ and $\mathbf{A}$ is of full row rank.

Geometrically speaking, the standard LP problem in (3) looks for the minimum of a linear function over a polyhedron whose edges are defined by a set of hyperplanes. Therefore, if there exists a finite solution for the LP problem, at least one of the intersection points (formally called basic solutions) of the hyperplanes should attain the minimum. For formal discussion of the optimality, a brief review of some terminologies in LP is provided. Let $\mathcal{N}$ denote the index set $\{1, \cdots, N\}$ of the unknowns, $\boldsymbol{z}$, in the LP problem in (3).

Definition 1 *A set $\mathcal{B}^* := \{B_1^*, \cdots, B_M^*\} \subset \mathcal{N}$ is called a basic index set, if $\mathbf{A}_{\mathcal{B}^*} := [\mathbf{A}_{B_1^*}, \cdots, \mathbf{A}_{B_M^*}]$ is invertible, where $\mathbf{A}_{B_i^*}$ is the B_i^*th column vector of $\mathbf{A}$ for $i = 1, \cdots, M$. $\mathbf{A}_{\mathcal{B}^*}$ is called the basic matrix associated with $\mathcal{B}^*$. Correspondingly, a vector $\boldsymbol{z}^* \in \mathcal{R}^N$ is called the basic solution associated with $\mathcal{B}^*$, if $\boldsymbol{z}^*$ satisfies*

$$\begin{cases} \boldsymbol{z}^*_{\mathcal{B}^*} := (z^*_{B_1^*}, \cdots, z^*_{B_M^*})' = \mathbf{A}^{-1}_{\mathcal{B}^*}\mathbf{b} \\ z_j^* = 0 \text{ for } j \in \mathcal{N} \setminus \mathcal{B}^*. \end{cases}$$

Definition 2 *Let $\boldsymbol{z}^*$ be the basic solution associated with $\mathcal{B}^*$.*

- *$\boldsymbol{z}^*$ is called a basic feasible solution if $\boldsymbol{z}^*_{\mathcal{B}^*} \geq \mathbf{0}$;*
- *$\boldsymbol{z}^*$ is called a non-degenerate basic feasible solution if $\boldsymbol{z}^*_{\mathcal{B}^*} > \mathbf{0}$;*
- *$\boldsymbol{z}^*$ is called a degenerate basic feasible solution if $\boldsymbol{z}^*_{\mathcal{B}^*} \geq \mathbf{0}$ and $z^*_{B_i^*} = 0$ for some $i \in \mathcal{M} := \{1, \cdots, M\}$;*
- *$\boldsymbol{z}^*$ is called an optimal basic solution if $\boldsymbol{z}^*$ is a solution of the LP problem.*

Since each basic solution is associated with its basic index set, the optimal basic solution can be identified with the optimal basic index set as defined below.

Definition 3 *A basic index set $\mathcal{B}^*$ is called a feasible basic index set if $\mathbf{A}^{-1}_{\mathcal{B}^*}\mathbf{b} \geq \mathbf{0}$. A feasible basic index set $\mathcal{B}^*$ is also called an optimal basic index set if*

$$\left[\mathbf{c} - \mathbf{A}' \left(\mathbf{A}^{-1}_{\mathcal{B}^*}\right)' \mathbf{c}_{\mathcal{B}^*}\right] \geq \mathbf{0}.$$

The following theorem indicates that the standard LP problem can be solved by finding the optimal basic index set.

Theorem 4 *For the LP problem in (3), let $\boldsymbol{z}^*$ be the basic solution associated with $\mathcal{B}^*$, an optimal basic index set. Then $\boldsymbol{z}^*$ is an optimal basic solution.*

Proof We need to show $\mathbf{c}'\boldsymbol{z} \geq \mathbf{c}'\boldsymbol{z}^*$ or $\mathbf{c}'(\boldsymbol{z} - \boldsymbol{z}^*) \geq \mathbf{0}$ for any feasible vector $\boldsymbol{z} \in \mathcal{R}^N$ with $\mathbf{A}\boldsymbol{z} = \mathbf{b}$ and $\boldsymbol{z} \geq 0$. Set $\boldsymbol{d} := (d_1, \cdots, d_N) := (\boldsymbol{z} - \boldsymbol{z}^*)$. From

$$\mathbf{A}\boldsymbol{d} = \mathbf{A}_{\mathcal{B}^*}\boldsymbol{d}_{\mathcal{B}^*} + \sum_{i \in \mathcal{N} \setminus \mathcal{B}^*} \mathbf{A}_i d_i = \mathbf{0},$$

we have

$$\boldsymbol{d}_{\mathcal{B}^*} = - \sum_{i \in \mathcal{N} \setminus \mathcal{B}^*} \mathbf{A}_{\mathcal{B}^*}^{-1}\mathbf{A}_i d_i.$$

Then,

$$\begin{aligned} \mathbf{c}'(\boldsymbol{z} - \boldsymbol{z}^*) &= \mathbf{c}'\boldsymbol{d} = \mathbf{c}_{\mathcal{B}^*}'\boldsymbol{d}_{\mathcal{B}^*} + \sum_{i \in \mathcal{N} \setminus \mathcal{B}^*} \mathrm{c}_i d_i \\ &= \sum_{i \in \mathcal{N} \setminus \mathcal{B}^*} (\mathrm{c}_i - \mathbf{c}_{\mathcal{B}^*}'\mathbf{A}_{\mathcal{B}^*}^{-1}\mathbf{A}_i) d_i. \end{aligned}$$

Recall that for $i \in \mathcal{N} \setminus \mathcal{B}^*$, $z_i^* = 0$, which implies $d_i := (z_i - z_i^*) \geq 0$. Together with $\left[\mathbf{c} - \mathbf{A}' \left(\mathbf{A}_{\mathcal{B}^*}^{-1}\right)' \mathbf{c}_{\mathcal{B}^*}\right] \geq \mathbf{0}$, it ensures $(\mathrm{c}_i - \mathbf{c}_{\mathcal{B}^*}'\mathbf{A}_{\mathcal{B}^*}^{-1}\mathbf{A}_i) d_i \geq 0$. Thus, we have $\mathbf{c}'\boldsymbol{d} \geq 0$. ■

2.2 Parametric Linear Programs

In practical applications, the cost coefficients $\mathbf{c}$ or the constraint constants $\mathbf{b}$ in (3) are often partially known or controllable so that they may be modeled linearly as $(\mathbf{c} + \lambda\mathbf{a})$ or $(\mathbf{b} + \omega\mathbf{b}^*)$ with some parameters λ and $\omega \in \mathcal{R}$, respectively. A family of regularization methods for feature selection to be discussed share this structure. Of main interest in this paper is generation of the solution path indexed by the control parameter λ (or ω) as it corresponds to a trajectory of possible models or prediction rules produced by each regularization method in the family. Although every parameter value creates a new LP problem in the setting, it is feasible to generate solutions for all values of the parameter via sequential updates. The new LP problems indexed by the parameters λ and ω are called the parametric-cost LP and parametric right-hand-side LP, respectively.

The standard form of a parametric-cost LP is defined as

$$\left\{ \begin{array}{ll} \min\limits_{\boldsymbol{z} \in \mathcal{R}^N} & (\mathbf{c} + \lambda\mathbf{a})'\boldsymbol{z} \\ \text{s.t.} & \mathbf{A}\boldsymbol{z} = \mathbf{b} \\ & \boldsymbol{z} \geq \mathbf{0}. \end{array} \right. \tag{4}$$

Since the basic index sets of the parametric-cost LP do not depend on the parameter λ, it is not hard to see that an optimal basic index set $\mathcal{B}^*$ for some fixed value of λ would remain optimal for a range of λ values, say, $[\underline{\lambda}, \overline{\lambda}]$. The interval is called the optimality interval of $\mathcal{B}^*$ for the parametric-cost LP problem. The following result adapted from Section 8.2.1 of Murty (1983) shows how to find the lower and upper bounds of the interval, given a fixed value of λ, say, λ^* and the associated optimal basic index set $\mathcal{B}^*$.

Corollary 5 *For a fixed $\lambda^* \geq 0$, let $\mathcal{B}^*$ be an optimal basic index set of the problem in (4) at $\lambda = \lambda^*$. Define*

$$\underline{\lambda} := \max_{\{j\,:\,\check{a}_j^* > 0; j \in \mathcal{N} \setminus \mathcal{B}^*\}} \left(-\frac{\check{c}_j^*}{\check{a}_j^*} \right) \tag{5}$$

$$\text{and} \quad \overline{\lambda} := \min_{\{j\,:\,\check{a}_j^* < 0; j \in \mathcal{N} \setminus \mathcal{B}^*\}} \left(-\frac{\check{c}_j^*}{\check{a}_j^*} \right),$$

where $\check{a}_j^* := a_j - \mathbf{a}'_{\mathcal{B}^*}\mathbf{A}_{\mathcal{B}^*}^{-1}\mathbf{A}_j$ and $\check{c}_j^* := c_j - \mathbf{c}'_{\mathcal{B}^*}\mathbf{A}_{\mathcal{B}^*}^{-1}\mathbf{A}_j$ for $j \in \mathcal{N}$. Then, $\mathcal{B}^*$ is an optimal basic index set of (4) for $\lambda \in [\underline{\lambda}, \overline{\lambda}]$, which includes λ^*.

Proof From the optimality of $\mathcal{B}^*$ for $\lambda = \lambda^*$, we have $\mathbf{A}_{\mathcal{B}^*}^{-1}\mathbf{b} \geq \mathbf{0}$ and

$$\left[\mathbf{c} - \mathbf{A}' \left(\mathbf{A}_{\mathcal{B}^*}^{-1}\right)' \mathbf{c}_{\mathcal{B}^*}\right] + \lambda^* \left[\mathbf{a} - \mathbf{A}' \left(\mathbf{A}_{\mathcal{B}^*}^{-1}\right)' \mathbf{a}_{\mathcal{B}^*}\right] \geq \mathbf{0},$$

which implies that $\check{c}_j^* + \lambda^* \check{a}_j^* \geq \mathbf{0}$ for $j \in \mathcal{N}$. To find the optimality interval $[\underline{\lambda}, \overline{\lambda}]$ of $\mathcal{B}^*$, by Theorem 4, we need to investigate the following inequality for each $j \in \mathcal{N}$:

$$\check{c}_j^* + \lambda \check{a}_j^* \geq 0. \tag{6}$$

It is easy to see that $\mathbf{A}_{\mathcal{B}^*}^{-1}\mathbf{A}_{B_i^*} = \mathbf{e}_i$ for $i \in \mathcal{M}$ since $\mathbf{A}_{B_i^*}$ is the ith column of $\mathbf{A}_{\mathcal{B}^*}$. Consequently, the jth entries of $(\mathbf{c}' - \mathbf{c}'_{\mathcal{B}^*}\mathbf{A}_{\mathcal{B}^*}^{-1}\mathbf{A})$ and $(\mathbf{a}' - \mathbf{a}'_{\mathcal{B}^*}\mathbf{A}_{\mathcal{B}^*}^{-1}\mathbf{A})$ are both 0 for $j \in \mathcal{B}^*$, and $\check{c}_j^* + \lambda \check{a}_j^* = 0$ for any λ. So, the inequality holds for any $\lambda \in \mathcal{R}$ and $j \in \mathcal{B}^*$. When $\check{a}_j^* > 0$ (or $\check{a}_j^* < 0$) for $j \in (\mathcal{N} \setminus \mathcal{B}^*)$, (6) holds if and only if $\lambda \geq -\check{c}_j^*/\check{a}_j^*$ (or $\lambda \leq -\check{c}_j^*/\check{a}_j^*$). Thus, the lower bound and the upper bound of the optimality interval of $\mathcal{B}^*$ are given by the $\underline{\lambda}$ and $\overline{\lambda}$ in (5). ■

Note that $\check{c}_j^*$ and $\check{a}_j^*$ define the relative cost coefficient of z_j. Since the number of basic index sets is finite for fixed $\mathbf{A}$, there exist only a finite number of optimal basic index sets of the problem in (4). Corollary 5 also implies that a version of the solution path of the problem as a function of λ, $\boldsymbol{z}(\lambda)$, is a step function.

On the other hand, if the parametric cost LP in (4) is recast in the form of (2), then the stepwise constant property of the solution path changes. The alternative complexity-bounded formulation of (4) is given by

$$\begin{cases} \min\limits_{\boldsymbol{z} \in \mathcal{R}^N, \delta \in \mathcal{R}} & \mathbf{c}'\boldsymbol{z} \\ \text{s.t.} & \mathbf{A}\boldsymbol{z} = \mathbf{b} \\ & \mathbf{a}'\boldsymbol{z} + \delta = s \\ & \boldsymbol{z} \geq \mathbf{0}, \delta \geq 0. \end{cases} \tag{7}$$

It can be transformed into a standard parametric right-hand-side LP problem:

$$\begin{cases} \min\limits_{\mathbb{Z} \in \mathcal{R}^{N+1}} & \mathbb{c}'\mathbb{Z} \\ \text{s.t.} & \mathbb{A}\mathbb{Z} = \mathbb{b} + \omega \mathbb{b}^* \\ & \mathbb{Z} \geq \mathbf{0} \end{cases} \tag{8}$$

by setting $\omega = s$, $\mathbb{Z} = \begin{bmatrix} \boldsymbol{z} \\ \delta \end{bmatrix}$, $\mathbb{c} = \begin{bmatrix} \mathbf{c} \\ 0 \end{bmatrix}$, $\mathbb{b} = \begin{bmatrix} \mathbf{b} \\ 0 \end{bmatrix}$, $\mathbb{b}^* = \begin{bmatrix} \mathbf{0} \\ 1 \end{bmatrix}$, and $\mathbb{A} = \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{a}' & 1 \end{bmatrix}$. Note that when $\mathbf{A}$ in (8) is of full row rank, so is $\mathbb{A}$. Let $\mathfrak{B}^*$ be an optimal basic index set of (8) at $\omega = \omega^*$. Similarly, we can show that $\mathfrak{B}^*$ is optimal for any ω satisfying $\mathbb{Z}_{\mathfrak{B}^*} = \mathbb{A}_{\mathfrak{B}^*}^{-1}(\mathbb{b} + \omega \mathbb{b}^*) \geq \mathbf{0}$, and there exist $\underline{\omega}$ and $\overline{\omega}$ such that $\mathfrak{B}^*$ is optimal for $\omega \in [\underline{\omega}, \overline{\omega}]$. This implies that a version of the solution path of (8) is a piecewise linear function.

3 Generating the Solution Path

Standard techniques for solving LP include simplex method, dual simplex method, tableau method, and interior point methods. Interior point methods (Karmarkar, 1984; Wright, 1992; Mehrotra, 1992; Wright, 1997) can be more effective in solving some large scale sparse LP problems than the simplex method. However, they are generally perceived to lack an efficient warm-starting scheme, which is important in generating the entire regularized solution path. For the reason, variants of simplex algorithms are considered in this paper.

Based on the basic concepts and the optimality condition of LP introduced in Section 2, we describe the simplex and tableau-simplex algorithms to generate the solution paths for (4) and (7). A similar treatment of the parametric LP can be found in Saaty and Gass (1954), and our description of the tableau-simplex algorithm is mainly adapted from Section 8.2 of Murty (1983).

Since the examples of the LP problem in Section 4 for feature selection involve non-negative $\mathbf{a}$, λ, and s only, we assume that they are non-negative in the following algorithms and take $s = 0$ (equivalently $\lambda = \infty$) as a starting value.

3.1 Simplex Algorithm

3.1.1 Initialization

Let $\boldsymbol{z}^0 := (z_1^0, \cdots, z_N^0)'$ denote the initial solution of (7) at $s = 0$. $\mathbf{a}'\boldsymbol{z}^0 = 0$ implies $z_j^0 = 0$ for all $j \notin \mathcal{I}_\mathbf{a} := \{i : \mathrm{a}_i = 0, i \in \mathcal{N}\}$. Thus, by extracting the coordinates of $\mathbf{c}$, $\boldsymbol{z}$, and the columns in $\mathbf{A}$ corresponding to $\mathcal{I}_\mathbf{a}$, we can simplify the initial LP problem of (4) and (7) to

$$\begin{cases} \min\limits_{\boldsymbol{z}_{\mathcal{I}_\mathbf{a}} \in \mathcal{R}^{|\mathcal{I}_\mathbf{a}|}} & \mathbf{c}_{\mathcal{I}_\mathbf{a}}{}'\boldsymbol{z}_{\mathcal{I}_\mathbf{a}} \\ \text{s.t.} & \mathbf{A}_{\mathcal{I}_\mathbf{a}}\boldsymbol{z}_{\mathcal{I}_\mathbf{a}} = \mathbf{b} \\ & \boldsymbol{z}_{\mathcal{I}_\mathbf{a}} \geq \mathbf{0} \end{cases}, \tag{9}$$

where $|\mathcal{I}_\mathbf{a}|$ is the cardinality of $\mathcal{I}_\mathbf{a}$. Accordingly, any initial optimal basic index set, $\mathcal{B}^0$ of (4) and (7) contains that of the reduced problem (9) and determines the initial solution $\boldsymbol{z}^0$.

3.1.2 Main Algorithm

For simplicity, we describe the algorithm for the solution path of the parametric-cost LP problem in (4) first, and then discuss how it also solves the complexity-bounded LP problem in (7).

Let $\mathcal{B}^l$ be the lth optimal basic index set at $\lambda = \lambda_{l-1}$. For convenience, define $\lambda_{-1} := \infty$, the starting value of the regularization parameter for the solution path of (4). Given $\mathcal{B}^l$, let $\boldsymbol{z}^l$ be the lth joint solution, which is given by $\boldsymbol{z}^l_{\mathcal{B}^l} = \mathbf{A}^{-1}_{\mathcal{B}^l}\mathbf{b}$ and $\boldsymbol{z}^l_j = 0$ for $j \in \mathcal{N} \setminus \mathcal{B}^l$. Since the optimal LP solution is identified by the optimal basic index set as in Theorem 4, it suffices to describe how to update the optimal basic index set as λ decreases. By the invertibility of the basic matrix associated with the index set, updating amounts to finding a new index that enters and the other that exits the current basic index set.

By Corollary 5, we can compute the lower bound of the optimality interval of $\mathcal{B}^l$ denoted by λ_l and identify the entry index associated with it. Let

$$j^l := \underset{\{j\,:\,\check{\mathrm{a}}^l_j > 0;\, j \in (\mathcal{N} \setminus \mathcal{B}^l)\}}{\arg\max} \left(-\frac{\check{\mathrm{c}}^l_j}{\check{\mathrm{a}}^l_j} \right), \tag{10}$$

where $\check{\mathrm{a}}^l_j := (\mathrm{a}_j - \mathbf{a}'_{\mathcal{B}^l}\mathbf{A}^{-1}_{\mathcal{B}^l}\mathbf{A}_j)$ and $\check{\mathrm{c}}^l_j := (\mathrm{c}_j - \mathbf{c}'_{\mathcal{B}^l}\mathbf{A}^{-1}_{\mathcal{B}^l}\mathbf{A}_j)$. Then, the lower bound is given by $\lambda_l := -\check{\mathrm{c}}^l_{j^l}/\check{\mathrm{a}}^l_{j^l}$, and $\mathcal{B}^l$ is optimal for $\lambda \in [\lambda_l, \lambda_{l-1}]$.

To determine the index exiting $\mathcal{B}^l$, consider the moving direction from $\boldsymbol{z}^l$ to the next joint solution. Define $\boldsymbol{d}^l := (d^l_1, \cdots, d^l_N)$ as

$$\begin{aligned} &\boldsymbol{d}^l_{\mathcal{B}^l} = -\mathbf{A}^{-1}_{\mathcal{B}^l}\mathbf{A}_{j^l}, d^l_{j^l} = 1, \text{ and} \\ &d^l_i = 0 \text{ for } i \in \mathcal{N} \setminus (\mathcal{B}^l \cup \{j^l\}). \end{aligned} \tag{11}$$

Lemma 13 in Appendix shows that $\boldsymbol{d}^l$ is the moving direction at $\lambda = \lambda_l$ in the sense that $\boldsymbol{z}^{l+1} = \boldsymbol{z}^l + \tau \boldsymbol{d}^l$ for some $\tau \geq 0$. For the feasibility of $\boldsymbol{z}^{l+1} \geq 0$, the step size τ can not exceed the minimum of $-z^l_i/d^l_i$ for $i \in \mathcal{B}^l$ with $d^l_i < 0$, and the index attaining the minimum is to leave $\mathcal{B}^l$. Denote the exit index by

$$i^l := \underset{i \in \{j:\, d^l_j < 0,\ j \in \mathcal{B}^l\}}{\arg\min} \left(-\frac{z^l_i}{d^l_i} \right). \tag{12}$$

Therefore, the optimal basic index set at $\lambda = \lambda_l$ is given by $\mathcal{B}^{l+1} := \mathcal{B}^l \cup \{j^l\} \setminus \{i^l\}$. More precisely, we can verify the optimality of $\mathcal{B}^{l+1}$ at $\lambda = \lambda_l$ by showing that

$$\begin{aligned} &(\mathbf{c} + \lambda_l \mathbf{a}) - \mathbf{A}' \left(\mathbf{A}^{-1}_{\mathcal{B}^{l+1}} \right)' (\mathbf{c}_{\mathcal{B}^{l+1}} + \lambda_l \mathbf{a}_{\mathcal{B}^{l+1}}) \\ = \quad &(\mathbf{c} + \lambda_l \mathbf{a}) - \mathbf{A}' \left(\mathbf{A}^{-1}_{\mathcal{B}^l} \right)' (\mathbf{c}_{\mathcal{B}^l} + \lambda_l \mathbf{a}_{\mathcal{B}^l}). \end{aligned} \tag{13}$$

The proof is given in Appendix B. Then the fact that $\mathcal{B}^l$ is optimal at $\lambda = \lambda_l$ implies that $\mathcal{B}^{l+1}$ is also optimal at $\lambda = \lambda_l$. As a result, the updating procedure can be repeated with $\mathcal{B}^{l+1}$ and λ_l successively until $\lambda_l < 0$ or equivalently $\check{\mathrm{c}}^l_{j^l} \geq 0$. The algorithm for updating the optimal basic index sets is summarized as follows.

---

1. Initialize the optimal basic index set at $\lambda_{-1} = \infty$ with $\mathcal{B}^0$.
2. Given $\mathcal{B}^l$, the lth optimal basic index set at $\lambda = \lambda_{l-1}$, determine the solution $\boldsymbol{z}^l$ by $\boldsymbol{z}^l_{\mathcal{B}^l} = \mathbf{A}^{-1}_{\mathcal{B}^l}\mathbf{b}$ and $\boldsymbol{z}^l_j = 0$ for $j \in \mathcal{N} \setminus \mathcal{B}^l$.
3. Find the entry index
$$j^l = \underset{j\,:\,\check{\mathrm{a}}^l_j > 0;\, j \in \mathcal{N} \setminus \mathcal{B}^l}{\arg\max} \left(-\frac{\check{\mathrm{c}}^l_j}{\check{\mathrm{a}}^l_j} \right).$$
4. Find the exit index
$$i^l = \underset{i \in \{j:\, d^l_j < 0,\ j \in \mathcal{B}^l\}}{\arg\min} \left(-\frac{z^l_i}{d^l_i} \right).$$
If there are multiple indices, choose one of them.
5. Update the optimal basic index set to $\mathcal{B}^{l+1} = \mathcal{B}^l \cup \{j^l\} \setminus \{i^l\}$ and λ_{l-1} to λ_l.
6. Terminate the algorithm if $\check{\mathrm{c}}^l_{j^l} \geq 0$ or equivalently $\lambda_l \leq 0$. Otherwise, increase l by 1 and repeat 2 – 5.

If $-z^l_{i^l}/d^l_{i^l} = 0$, then $\boldsymbol{z}^l = \boldsymbol{z}^{l+1}$, which may result in the problem of cycling among several basic index sets with the same solution. We defer the description of the tableau-simplex algorithm which can avoid the cycling problem to Section 3.2. For brevity, we just assume that $\boldsymbol{z}^l + \tau \boldsymbol{d}^l \geq \mathbf{0}$ for some $\tau > 0$ so that $\boldsymbol{z}^l \neq \boldsymbol{z}^{l+1}$ for each l and call this *non-degeneracy* assumption. Under this assumption, suppose the simplex algorithm terminates after J iterations with $\{(\boldsymbol{z}^l, \lambda_l) : l = 0, 1, \cdots, J\}$. Then the entire solution path is obtained as described below.

Theorem 6 *The solution path of (4) is*

$$\begin{cases} \boldsymbol{z}^0 & \text{for } \lambda > \lambda_0 \\ \boldsymbol{z}^l & \text{for } \lambda_l < \lambda < \lambda_{l-1},\ \ l = 1, \cdots, J \\ \tau \boldsymbol{z}^l + (1-\tau)\boldsymbol{z}^{l+1} & \text{for } \lambda = \lambda_l \text{ and } \tau \in [0,1],\ l = 0, \cdots, J-1. \end{cases} \tag{14}$$

Likewise, the solutions to the alternative formulation of (7) with the complexity bound can be obtained as a function of s. By the correspondence of the two formulations, the lth joint of the piecewise linear solution is given by $s_l = \mathbf{a}'\boldsymbol{z}^l$, and the solution between the joints is a linear combination of $\boldsymbol{z}^l$ and $\boldsymbol{z}^{l+1}$ as described in Theorem 7 below. Its proof is in Appendix C. To the best of our knowledge, this direct proof of the piecewise linearity of the solution to (7) in the parametric right-hand-side LP formulation is new.

Theorem 7 *For $s \geq 0$, the solution path of (7) can be expressed as*

$$\begin{cases} \frac{s_{l+1} - s}{s_{l+1} - s_l}\boldsymbol{z}^l + \frac{s - s_l}{s_{l+1} - s_l}\boldsymbol{z}^{l+1} & \text{if } s_l \leq s < s_{l+1} \text{ and } l = 0, \cdots, J-1 \\ \boldsymbol{z}^J & \text{if } s \geq s_J. \end{cases}$$

Notice that when indexed by λ, the solutions at the joints λ_l are not unique, but when parametrized by s, the solution path is expressed uniquely as a piecewise linear function of s by tracing those line segments of two consecutive joint solutions. In essence, the solutions generated by the simplex algorithm are *indexed by the optimal basic index sets* $\mathcal{B}^l$, and the sequences of λ_l and s_l are completely determined by $\mathcal{B}^l$ as a result. Hence, to the extent that the optimal basic index sets are uniquely determined, the regularized solution path is defined uniquely.

3.2 Tableau-Simplex Algorithm

The non-degeneracy assumption in the simplex method that any two consecutive joint solutions are different may not hold in practice for many problems. When some columns of a basic matrix are discrete, the assumption may fail at some degenerate joint solutions. To deal with more general settings where the cycling problem may occur in generating the LP solution path by the simplex method, we discuss the *tableau-simplex* algorithm.

A tableau refers to a matrix which contains all the information about the LP. It consists of the relevant terms in LP associated with a basic matrix such as the basic solution and the cost.

Definition 8 *For a basic index set $\mathcal{B}^*$, its tableau is defined as*

	zeroth column	*pivot columns*
cost row	$-\mathbf{c}'_{\mathcal{B}^*}\mathbf{A}^{-1}_{\mathcal{B}^*}\mathbf{b}$	$\mathbf{c}' - \mathbf{c}'_{\mathcal{B}^*}\mathbf{A}^{-1}_{\mathcal{B}^*}\mathbf{A}$
penalty row	$-\mathbf{a}'_{\mathcal{B}^*}\mathbf{A}^{-1}_{\mathcal{B}^*}\mathbf{b}$	$\mathbf{a}' - \mathbf{a}'_{\mathcal{B}^*}\mathbf{A}^{-1}_{\mathcal{B}^*}\mathbf{A}$
pivot rows	$\mathbf{A}^{-1}_{\mathcal{B}^*}\mathbf{b}$	$\mathbf{A}^{-1}_{\mathcal{B}^*}\mathbf{A}$

We follow the convention for the names of the columns and rows in the tableau. For reference, see Murty (1983) and Bertsimas and Tsitsiklis (1997). Note that the zeroth column contains $\boldsymbol{z}^*_{\mathcal{B}^*} := \mathbf{A}^{-1}_{\mathcal{B}^*}\mathbf{b}$, the non-zero part of the basic solution, $-\mathbf{c}'_{\mathcal{B}^*}\mathbf{A}^{-1}_{\mathcal{B}^*}\mathbf{b} = -\mathbf{c}'\boldsymbol{z}^*$, the negative cost, and $-\mathbf{a}'_{\mathcal{B}^*}\mathbf{A}^{-1}_{\mathcal{B}^*}\mathbf{b} = -\mathbf{a}'\boldsymbol{z}^*$, the negative penalty of $\boldsymbol{z}^*$ associated with $\mathcal{B}^*$, and the pivot columns contain $\check{\mathrm{c}}^*_j$'s and $\check{\mathrm{a}}^*_j$'s. The algorithm to be discussed updates the basic index sets by using the tableau, in particular, by ordering some rows of the tableau. To describe the algorithm, we introduce the lexicographic order of vectors first.

Definition 9 *For $\boldsymbol{v}$ and $\boldsymbol{w} \in \mathcal{R}^n$, we say that $\boldsymbol{v}$ is lexicographically greater than $\boldsymbol{w}$ (denoted by $\boldsymbol{v} \overset{L}{>} \boldsymbol{w}$) if the first non-zero entry of $\boldsymbol{v} - \boldsymbol{w}$ is strictly positive. We say that $\boldsymbol{v}$ is lexicographically positive if $\boldsymbol{v} \overset{L}{>} 0$.*

Now, consider the parametric-cost LP in (4).

3.2.1 Initial Tableau

With the index set $\mathcal{B}^0$, initialize the tableau. Since $\boldsymbol{z}^0_{\mathcal{B}^0} = \mathbf{A}^{-1}_{\mathcal{B}^0}\mathbf{b} \geq 0$ and the columns of $\mathbf{A}$ can be rearranged such that the submatrix with the first M columns of $\mathbf{A}^{-1}_{\mathcal{B}^0}\mathbf{A}$ is $\mathbf{I}$, we assume that the pivot rows, $[\mathbf{A}^{-1}_{\mathcal{B}^0}\mathbf{b} \;\; \mathbf{A}^{-1}_{\mathcal{B}^0}\mathbf{A}]$, of the initial tableau are lexicographically positive. In other words, there is a permutation $\pi : \mathcal{N} \to \mathcal{N}$ which maps $\mathcal{B}^0$ to $\mathcal{M} := \{1, \cdots, M\}$, and we can replace the problem with the π-permuted version (e.g., $\boldsymbol{z}_{\pi(\mathcal{N})}$ and $\mathbf{A}_{\pi(\mathcal{N})}$).

3.2.2 Updating Tableau

Given the current optimal basic index set $\mathcal{B}^l$, the current tableau is

	zeroth column	pivot columns
cost row	$-\mathbf{c}'_{\mathcal{B}^l}\mathbf{A}^{-1}_{\mathcal{B}^l}\mathbf{b}$	$\mathbf{c}' - \mathbf{c}'_{\mathcal{B}^l}\mathbf{A}^{-1}_{\mathcal{B}^l}\mathbf{A}$
penalty row	$-\mathbf{a}'_{\mathcal{B}^l}\mathbf{A}^{-1}_{\mathcal{B}^l}\mathbf{b}$	$\mathbf{a}' - \mathbf{a}'_{\mathcal{B}^l}\mathbf{A}^{-1}_{\mathcal{B}^l}\mathbf{A}$
pivot rows	$\mathbf{A}^{-1}_{\mathcal{B}^l}\mathbf{b}$	$\mathbf{A}^{-1}_{\mathcal{B}^l}\mathbf{A}$

Suppose all the pivot rows of the current tableau are lexicographically positive. The tableau-simplex algorithm differs from the simplex algorithm only in the way the exit index is determined. The following procedure is generalization of Step 4 in the simplex algorithm for finding the exit index.

Step 4. Let $\boldsymbol{u}^l := (u^l_1, \cdots, u^l_M)' := \mathbf{A}^{-1}_{\mathcal{B}^l}\mathbf{A}_{j^l}$. For each $i \in \mathcal{M}$ with $u^l_i > 0$, divide the ith pivot row (including the entry in the zeroth column) by u^l_i. And, among those rows, find the index, i^l_*, of the lexicographically smallest row. Then, $i^l := B^l_{i^l_*}$ is the exit index.

Remark 10 *Since $\boldsymbol{u}^l = -\boldsymbol{d}^l_{\mathcal{B}^l}$, if i^l in (12) is unique with $z^l_{i^l} > 0$, then it is the same as the lexicographically smallest row that the tableau-simplex algorithm seeks. Hence the two algorithms coincide. The simplex algorithm determines the exit index based only on the zeroth column in the tableau while the lexicographic ordering involves the pivot columns additionally. The optimality of $\mathcal{B}^l$ for $\lambda \in [\lambda_l, \lambda_{l-1}]$ immediately follows by the same Step 3, and (13) remains to hold true for the exit index i^l of the tableau-simplex algorithm, which implies the optimality of $\mathcal{B}^{l+1}$ at $\lambda = \lambda_l$.*

Some characteristics of the updated tableau associated with $\mathcal{B}^{l+1}$ are described in the next theorem. The proof is adapted from that for the lexicographic pivoting rule in Bertsimas and Tsitsiklis (1997) p. 108–111. See Appendix D for details.

Theorem 11 *For the updated basic index set $\mathcal{B}^{l+1}$ by the tableau-simplex algorithm,*

i) all the pivot rows of the updated tableau are still lexicographically positive, and

ii) the updated cost row is lexicographically greater than that for $\mathcal{B}^l$.

Since $\mathbf{A}_{\mathcal{B}^{l+1}}^{-1}\mathbf{b}$ is the 'zeroth column' of the pivot rows, i) says that the basic solution for $\mathcal{B}^{l+1}$ is feasible, i.e., $\boldsymbol{z}^{l+1} \geq \mathbf{0}$. Moreover, it implies that the updating procedure can be repeated with $\mathcal{B}^{l+1}$ and the new tableau.

It is not hard to see that $\boldsymbol{z}^{l+1} = \boldsymbol{z}^l$ if and only if $z_{i^l}^l = 0$ (see the proof of Theorem 11 in the Appendix for more details). When $z_{i^l}^l = 0$, $\boldsymbol{z}^{l+1} = \boldsymbol{z}^l$, however the tableau-simplex algorithm uniquely updates $\mathcal{B}^{l+1}$ such that the previous optimal basic index sets $\mathcal{B}^l$'s never reappear in the process. This anti-cycling property is guaranteed by ii). By ii), we can strictly order the optimal basic index sets $\mathcal{B}^l$ based on their cost rows. Because of this and the fact that all possible basic index sets are finite, the total number of iterations must be finite. This proves the following.

Corollary 12 *The tableau updating procedure terminates after a finite number of iterations.*

Suppose that the tableau-simplex algorithm stops after J iterations with $\lambda_J \leq 0$. In parallel to the simplex algorithm, the tableau-simplex algorithm outputs the sequence $\{(\boldsymbol{z}^l, s_l, \lambda_l) : l = 0, \cdots, J\}$, and the solution paths for (4) and (7) admit the same forms as in Theorem 6 and Theorem 7 except for any duplicate joints λ_l and s_l.

4 Examples of LP for Regularization

We provide several concrete examples of LP problems that arise in statistics for feature selection via regularization. For each example, we identify the elements in the standard LP form, and discuss their commonalities across different examples and how they can be utilized for efficient computation.

4.1 l_1-Norm Quantile Regression

Quantile regression is a regression technique, introduced by Koenker and Bassett (1978), intended to estimate conditional quantile functions. It is obtained by replacing the squared error loss of the classical linear regression for the conditional mean function with a piecewise linear loss called the check function. For a general introduction to quantile regression, see Koenker and Hallock (2001).

For simplicity, assume that the conditional quantiles are linear in the predictors. Given a data set, $\{(\mathbf{x}_i, y_i) : \mathbf{x}_i \in \mathcal{R}^p, y_i \in \mathcal{R}, i = 1, \cdots, n\}$, the τth conditional quantile function is estimated by

$$\min_{\beta_0 \in \mathcal{R}, \boldsymbol{\beta} \in \mathcal{R}^p} \quad \sum_{i=1}^n \rho_\tau \left(y_i - \beta_0 - \mathbf{x}_i\boldsymbol{\beta}\right), \tag{15}$$

where $\boldsymbol{\beta}_0$ and $\boldsymbol{\beta} := (\beta_1, \ldots, \beta_p)'$ are the quantile regression coefficients for $\tau \in (0, 1)$, and $\rho_\tau(\cdot)$ is the check function defined as

$$\rho_\tau(t) := \begin{cases} \tau \cdot t & \text{for } t > 0 \\ -(1-\tau) \cdot t & \text{for } t \leq 0. \end{cases}$$

For example, when $\tau = 1/2$, the median regression function is estimated. The standard quantile regression problem in (15) can be cast as an LP problem itself. Barrodale and Roberts (1973) propose an improved

tableau-simplex algorithm for median regression. Koenker and D'Orey (1987) modify the algorithm to process quantile regression. Koenker and D'Orey (1994) further generalize the algorithm for enumeration of the entire range of quantile functions parametrized by τ, treating it as a parametric cost LP problem. Since the problem is somewhat different from an array of statistical optimization problems for feature selection that we intend to address in this paper, we skip discussion of the topic and refer the readers to Koenker and D'Orey (1994) and Koenker (2005).

Aiming to estimate the conditional quantile function simultaneously with selection of relevant predictors, Li and Zhu (2008) propose the l_1-norm quantile regression. It is defined by the following constrained optimization problem:

$$\begin{cases} \min\limits_{\beta_0 \in \mathcal{R},\, \boldsymbol{\beta} \in \mathcal{R}^p} & \sum_{i=1}^n \rho_\tau \left(y_i - \beta_0 - \mathbf{x}_i \boldsymbol{\beta}\right) \\ \text{s.t.} & \|\boldsymbol{\beta}\|_1 \leq s, \end{cases}$$

where $s > 0$ is a regularization parameter. Equivalently, with another tuning parameter λ, the l_1-norm quantile regression can be recast as

$$\begin{cases} \min\limits_{\beta_0 \in \mathcal{R},\, \boldsymbol{\beta} \in \mathcal{R}^p,\, \boldsymbol{\zeta} \in \mathcal{R}^n} & \sum_{i=1}^n \{\tau(\zeta_i)_+ + (1-\tau)(\zeta_i)_-\} + \lambda \|\boldsymbol{\beta}\|_1 \\ \text{s.t.} & \beta_0 + \mathbf{x}_i \boldsymbol{\beta} + \zeta_i = y_i \text{ for } i = 1, \cdots, n, \end{cases} \tag{16}$$

where $(x)_+ = \max(x, 0)$ and $(x)_- = \max(-x, 0)$. Now it is straightforward to formulate (16) as an LP parametrized by λ, which is a common feature of the examples discussed in this section. For the non-negativity constraint in the standard form of LP, consider both positive and negative parts of each variable and denote, for example, $((\beta_1)_+, \ldots, (\beta_p)_+)'$ by $\boldsymbol{\beta}^+$ and $((\beta_1)_-, \ldots, (\beta_p)_-)'$ by $\boldsymbol{\beta}^-$. Note that $\boldsymbol{\beta} = \boldsymbol{\beta}^+ - \boldsymbol{\beta}^-$ and the l_1-norm $\|\boldsymbol{\beta}\|_1 := \sum_{i=1}^p |\beta_i|$ is given by $\mathbf{1}'(\boldsymbol{\beta}^+ + \boldsymbol{\beta}^-)$ with $\mathbf{1} := (1, \cdots, 1)'$ of appropriate length. Let $\boldsymbol{Y} := (y_1, \cdots, y_n)'$, $\mathbf{X} := (\mathbf{x}_1', \cdots, \mathbf{x}_n')'$, $\boldsymbol{\zeta} := (\zeta_1, \cdots, \zeta_n)'$, and $\mathbf{0} := (0, \cdots, 0)'$ of appropriate length. Then the following elements define the l_1-norm quantile regression in the standard form of a parametric-cost LP in (4):

$$\begin{array}{rccccccc} \boldsymbol{z} := & (& \beta_0^+ & \beta_0^- & (\boldsymbol{\beta}^+)' & (\boldsymbol{\beta}^-)' & (\boldsymbol{\zeta}^+)' & (\boldsymbol{\zeta}^-)' &)' \\ \mathbf{c} := & (& 0 & 0 & \mathbf{0}' & \mathbf{0}' & \tau\mathbf{1}' & (1-\tau)\mathbf{1}' &)' \\ \mathbf{a} := & (& 0 & 0 & \mathbf{1}' & \mathbf{1}' & \mathbf{0}' & \mathbf{0}' &)' \\ \mathbf{A} := & (& \mathbf{1} & -\mathbf{1} & \mathbf{X} & -\mathbf{X} & \mathbf{I} & -\mathbf{I} &) \\ \mathbf{b} := & \boldsymbol{Y} & & & & & & & \end{array}$$

with a total of $N = 2(1 + p + n)$ variables and $M = n$ equality constraints.

4.2 l_1-Norm Support Vector Machine

Consider a binary classification problem where $y_i \in \{-1, 1\}, i = 1, \cdots, n$ denote the class labels. The Support Vector Machine (SVM) introduced by Cortes and Vapnik (1995) is a classification method that finds the optimal hyperplane maximizing the margin between the classes. It is another example of a regularization method with a margin-based hinge loss and the ridge regression type l_2 norm penalty. The optimal hyperplane ($\beta_0 + \mathbf{x}\boldsymbol{\beta} = 0$) in the standard SVM is determined by the solution to the problem:

$$\min_{\beta_0 \in \mathcal{R},\, \boldsymbol{\beta} \in \mathcal{R}^p} \sum_{i=1}^n \{1 - y_i \left(\beta_0 + \mathbf{x}_i \boldsymbol{\beta}\right)\}_+ + \lambda \|\boldsymbol{\beta}\|_2^2,$$

where $\lambda > 0$ is a tuning parameter. Replacing the l_2 norm with the l_1 norm for selection of variables, Bradley and Mangasarian (1998) and Zhu et al. (2004) arrive at a variant of the soft-margin SVM:

$$\left\{ \begin{array}{cl} \min\limits_{\beta_0 \in \mathcal{R}, \boldsymbol{\beta} \in \mathcal{R}^p, \boldsymbol{\zeta} \in \mathcal{R}^n} & \sum_{i=1}^n (\zeta_i)_+ + \lambda \|\boldsymbol{\beta}\|_1 \\ \text{s.t.} & y_i(\beta_0 + \mathbf{x}_i \boldsymbol{\beta}) + \zeta_i = 1 \text{ for } i = 1, \cdots, n. \end{array} \right. \tag{17}$$

Similarly, this l_1-norm SVM can be formulated as a parametric cost LP with the following elements in the standard form:

$$\begin{array}{rccccccccc} \boldsymbol{z} := & (& \beta_0^+ & \beta_0^- & (\boldsymbol{\beta}^+)' & (\boldsymbol{\beta}^-)' & (\boldsymbol{\zeta}^+)' & (\boldsymbol{\zeta}^-)' &)' \\ \mathbf{c} := & (& 0 & 0 & \mathbf{0}' & \mathbf{0}' & \mathbf{1}' & \mathbf{0}' &)' \\ \mathbf{a} := & (& 0 & 0 & \mathbf{1}' & \mathbf{1}' & \mathbf{0}' & \mathbf{0}' &)' \\ \mathbf{A} := & (& \boldsymbol{Y} & -\boldsymbol{Y} & \text{diag}(\boldsymbol{Y})\mathbf{X} & -\text{diag}(\boldsymbol{Y})\mathbf{X} & \mathbf{I} & -\mathbf{I} &) \\ \mathbf{b} := & \mathbf{1}. & & & & & & & \end{array}$$

This example will be revisited in great detail in Section 5.

4.3 l_1-Norm Functional Component Selection

We have considered only linear functions in the original variables for conditional quantiles and separating hyperplanes so far. In general, the technique of l_1 norm regularization for variable selection can be extended to nonparametric regression and classification. Although many different extensions are possible, we discuss here a specific extension for feature selection which is well suited to a wide range of function estimation and prediction problems. In a nutshell, the space of linear functions is substituted with a rich function space such as a reproducing kernel Hilbert space (Wahba, 1990; Schölkopf and Smola, 2002) where functions are decomposed of interpretable functional components, and the decomposition corresponds to a set of different kernels which generate the functional subspaces. Let an ANOVA-like decomposition of f with, say, d components be $f = f_1 + \cdots + f_d$ and $K_\nu, \nu = 1, \ldots, d$ be the associated kernels. Non-negative weights θ_ν are then introduced for recalibration of the functional components f_ν. Treating f_ν's as features and restricting the l_1 norm of $\boldsymbol{\theta} := (\theta_1, \ldots, \theta_d)'$ akin to the LASSO leads to a general procedure for feature selection and shrinkage. Detailed discussions of the idea can be found in Lin and Zhang (2006); Gunn and Kandola (2002); Zhang (2006); Lee et al. (2006). More generally, Micchelli and Pontil (2005) treat it as a regularization procedure for optimal kernel combination.

For illustration, we consider the "θ-step" of the structured SVM in Lee et al. (2006), which yields another parametric cost LP problem. For generality, consider a k-category problem with potentially different misclassification costs. The class labels are coded by k-vectors; $y_i = (y_i^1, \ldots, y_i^k)'$ denotes a vector with $y_i^j = 1$ and $-1/(k-1)$ elsewhere if the ith observation falls into class j. $L(y_i) = (L_{y_i}^1, \ldots, L_{y_i}^k)$ is a misclassification cost vector, where $L_j^{j'}$ is the cost of misclassifying j as j'. The SVM aims to find $f = (f^1, \ldots, f^k)'$ closely matching an appropriate class code y given $\mathbf{x}$ which induces a classifier $\phi(\mathbf{x}) = \arg\max_{j=1,\ldots,k} f^j(\mathbf{x})$. Suppose that each f^j is of the form $\beta_0^j + h^j(\mathbf{x}) := \beta_0^j + \sum_{i=1}^n \beta_i^j \sum_{\nu=1}^d \theta_\nu K_\nu(\mathbf{x}_i, \mathbf{x})$. Define the squared norm of h^j as $\|h^j\|_K^2 := (\boldsymbol{\beta}^j)' \left(\sum_{\nu=1}^d \theta_\nu \mathcal{K}_\nu \right) \boldsymbol{\beta}^j$, where $\boldsymbol{\beta}^j := (\beta_1^j, \ldots, \beta_n^j)'$ is the jth coefficient vector, and $\mathcal{K}_\nu$ is the n by n kernel matrix associated with K_ν. With the extended hinge loss $\mathcal{L}\{y_i, f(\mathbf{x}_i)\} := L(y_i)\{f(\mathbf{x}_i) - y_i\}_+$, the structured SVM finds f with $\boldsymbol{\beta}$ and $\boldsymbol{\theta}$ minimizing

$$\sum_{i=1}^n L(y_i)\{f(\mathbf{x}_i) - y_i\}_+ + \frac{\lambda}{2} \sum_{j=1}^k \|h^j\|_K^2 + \lambda_\theta \sum_{\nu=1}^d \theta_\nu \tag{18}$$

subject to $\theta_\nu \geq 0$ for $\nu = 1, \ldots, d$. λ and λ_θ are tuning parameters. By alternating estimation of $\boldsymbol{\beta}$ and $\boldsymbol{\theta}$, we attempt to find the optimal kernel configuration (a linear combination of pre-specified kernels) and the coefficients associated with the optimal kernel. The θ-step refers to optimization of the functional component weights $\boldsymbol{\theta}$ given $\boldsymbol{\beta}$. More specifically, treating $\boldsymbol{\beta}$ as fixed, the weights of the features are chosen to minimize

$$\sum_{j=1}^{k} (\mathbf{L}^j)' \left\{ \beta_0^j \mathbf{1} + \sum_{\nu=1}^{d} \theta_\nu \mathcal{K}_\nu \boldsymbol{\beta}^j - \boldsymbol{y}^j \right\}_+ + \frac{\lambda}{2} \sum_{j=1}^{k} (\boldsymbol{\beta}^j)' \left(\sum_{\nu=1}^{d} \theta_\nu \mathcal{K}_\nu \right) \boldsymbol{\beta}^j + \lambda_\theta \sum_{\nu=1}^{d} \theta_\nu,$$

where $\mathbf{L}^j := (L^j_{y_1}, \ldots, L^j_{y_n})'$ and $\boldsymbol{y}^j = (y_1^j, \ldots, y_n^j)'$.

This optimization problem can be rephrased as

$$\begin{cases} \min\limits_{\boldsymbol{\zeta} \in \mathcal{R}^{nk}, \boldsymbol{\theta} \in \mathcal{R}^d} & \sum_{j=1}^{k} (\mathbf{L}^j)'(\boldsymbol{\zeta}^j)_+ + \frac{\lambda}{2} \sum_{\nu=1}^{d} \theta_\nu \left(\sum_{j=1}^{k} (\boldsymbol{\beta}^j)' \mathcal{K}_\nu \boldsymbol{\beta}^j \right) + \lambda_\theta \sum_{\nu=1}^{d} \theta_\nu \\ \text{s.t.} & \sum_{\nu=1}^{d} \theta_\nu \mathcal{K}_\nu \boldsymbol{\beta}^j - \boldsymbol{\zeta}^j = \boldsymbol{y}^j - \beta_0^j \mathbf{1} \text{ for } j = 1, \ldots, k \\ & \theta_\nu \geq 0 \text{ for } \nu = 1, \ldots, d. \end{cases}$$

Let $\mathrm{g}_\nu := (\lambda/2) \sum_{j=1}^{k} (\boldsymbol{\beta}^j)' \mathcal{K}_\nu \boldsymbol{\beta}^j$, $\mathbf{g} := (\mathrm{g}_1, \cdots, \mathrm{g}_d)'$, $\mathbf{L} := \left((\mathbf{L}^1)', \cdots, (\mathbf{L}^k)' \right)'$, and $\boldsymbol{\zeta} := \left((\boldsymbol{\zeta}^1)', \cdots, (\boldsymbol{\zeta}^k)' \right)'$. Also, let

$$\mathbf{X} := \begin{bmatrix} \mathcal{K}_1 \boldsymbol{\beta}^1 & \cdots & \mathcal{K}_d \boldsymbol{\beta}^1 \\ \vdots & \ddots & \vdots \\ \mathcal{K}_1 \boldsymbol{\beta}^k & \cdots & \mathcal{K}_d \boldsymbol{\beta}^k \end{bmatrix}.$$

Then the following elements define the θ-step as a parametric cost LP indexed by λ_θ with $N = (d + 2nk)$ variables and $M = nk$ equality constraints:

$$\begin{aligned} \boldsymbol{z} &:= \begin{pmatrix} \boldsymbol{\theta}' & (\boldsymbol{\zeta}^+)' & (\boldsymbol{\zeta}^-)' \end{pmatrix}' \\ \mathbf{c} &:= \begin{pmatrix} \mathbf{g}' & \mathbf{L}' & \mathbf{0}' \end{pmatrix}' \\ \mathbf{a} &:= \begin{pmatrix} \mathbf{1}' & \mathbf{0}' & \mathbf{0}' \end{pmatrix}' \\ \mathbf{A} &:= \begin{pmatrix} \mathbf{X} & -\mathbf{I} & \mathbf{I} \end{pmatrix} \\ \mathbf{b} &:= ((\boldsymbol{y}^1 - \beta_0^1 \mathbf{1})', \ldots, (\boldsymbol{y}^k - \beta_0^k \mathbf{1})')'. \end{aligned}$$

4.4 Regularization for Grouping or Clustering of Features via LP

In many real applications, covariates are often grouped in nature, where group selection may be more pertinent than individual variable or feature selection. For example, a set of dummy variables created for a categorical variable or a factor form a natural group.

For description of grouped regularization, consider a standard linear model with J groups of variables:

$$\boldsymbol{Y} = \beta_0 + \sum_{j=1}^{J} \mathbf{X}_j \boldsymbol{\beta}_j + \boldsymbol{\epsilon},$$

where $\boldsymbol{Y}$ and $\boldsymbol{\epsilon}$ are n-vectors, $\mathbf{X}_j$ is an $n \times p_j$ matrix associated with the jth group of variables, and $\boldsymbol{\beta}_j := (\beta_{1j}, \cdots, \beta_{p_j j})'$ is a coefficient vector of size p_j for $j = 1, \cdots, J$. Let $\boldsymbol{\beta} := (\boldsymbol{\beta}_1', \cdots, \boldsymbol{\beta}_J')'$ and

$\mathbf{X} := (\mathbf{X}_1, \cdots, \mathbf{X}_J)$. For selection of important variable groups and estimation of the corresponding $\boldsymbol{\beta}$, Yuan and Lin (2006) propose a grouped LASSO penalty defined as

$$\|\boldsymbol{\beta}\|_{Glasso} := \sum_{j=1}^{J} \|\boldsymbol{\beta}_j\|_2$$

in the regression context. However, computationally different from the ordinary LASSO with ℓ_1 norm penalty (Tibshirani, 1996), the solution $\boldsymbol{\beta}$ of the grouped LASSO is not piecewise linear in the regularization parameter λ, and thus, it has to be calculated at each λ in general.

For easier computation and complete enumeration of the solution by piecewise linearity, one may consider an alternative penalty for grouped variable selection defined via the sup-norm:

$$\|\boldsymbol{\beta}\|_{F_\infty} := \sum_{j=1}^{J} \|\boldsymbol{\beta}_j\|_\infty, \tag{19}$$

which is suggested by Zou and Yuan (2008) originally for the SVM and named the *factorwise infinity* norm penalty.

As another variant, by noticing that the sup-norm penalty tends to equalize coefficients, Bondell and Reich (2008) propose the so-called OSCAR (Octagonal Shrinkage and Clustering Algorithm for Regression) penalty. It combines the ℓ_1 norm and the sup norm for simultaneous selection and clustering of correlated predictors which have a similar effect on the response. The OSCAR penalty for $\boldsymbol{\beta} := (\beta_1, \ldots, \beta_p)'$ is given by

$$\|\boldsymbol{\beta}\|_{Oscar} := \sum_{1 \le j \le k \le p} \max\{|\beta_j|, |\beta_k|\}. \tag{20}$$

The penalties for grouped regularization and clustering in (19) and (20) are of linear nature. When combined with piecewise linear loss functions given in the previous subsections, they also produce parametric LP problems. Hence, the algorithms in Section 3 are readily applicable.

For example, grouped median regression with the F_∞ norm penalty in (19) finds the coefficients, β_0 and $\boldsymbol{\beta}$ that minimize

$$\sum_{i=1}^{n} |y_i - \beta_0 - \sum_{j=1}^{J} \mathbf{x}_{ij}\boldsymbol{\beta}_j| + \lambda \|\boldsymbol{\beta}\|_{F_\infty}. \tag{21}$$

By introducing non-negative slack variables $\boldsymbol{\zeta}^+$, $\boldsymbol{\zeta}^-$, $\boldsymbol{\rho}^+ := (\rho_1^+, \cdots, \rho_J^+)'$, and $\boldsymbol{\eta}^+ := (\boldsymbol{\eta}_1^{+\prime}, \cdots, \boldsymbol{\eta}_J^{+\prime})'$, which are defined through the following relations:

$$\boldsymbol{\zeta} := \boldsymbol{\zeta}^+ - \boldsymbol{\zeta}^- := \boldsymbol{Y} - \beta_0 - \sum_{j=1}^{J} \mathbf{X}_j\boldsymbol{\beta}_j,$$
$$\rho_j^+ \mathbf{1} = \boldsymbol{\beta}_j^+ + \boldsymbol{\beta}_j^- + \boldsymbol{\eta}_j^+ \text{ with } \boldsymbol{\beta}_j = \boldsymbol{\beta}_j^+ - \boldsymbol{\beta}_j^- \text{ for } j = 1, \cdots, J,$$

the optimization problem in (21) can be formulated as a parametric LP:

$$\begin{cases} \min & \mathbf{1}'(\boldsymbol{\zeta}^+ + \boldsymbol{\zeta}^-) \\ \text{s.t.} & \boldsymbol{\zeta}^+ - \boldsymbol{\zeta}^- = \boldsymbol{Y} - (\beta_0^+ - \beta_0^-) - \sum_{j=1}^{J} \mathbf{X}_j(\boldsymbol{\beta}_j^+ - \boldsymbol{\beta}_j^-) \\ & (\rho_1^+ \mathbf{1}_{p_1}', \cdots, \rho_J^+ \mathbf{1}_{p_J}')' = \boldsymbol{\beta}^+ + \boldsymbol{\beta}^- + \boldsymbol{\eta}^+ \\ & \boldsymbol{\zeta}^+, \boldsymbol{\zeta}^-, \boldsymbol{\rho}^+, \boldsymbol{\eta}^+, \boldsymbol{\beta}^+, \boldsymbol{\beta}^- \ge \mathbf{0}. \end{cases}$$

In the standard form of (4), it has

$$
\begin{array}{rl}
\boldsymbol{z} := & (\ (\boldsymbol{\zeta}^+)' \quad (\boldsymbol{\eta}^+)' \quad (\boldsymbol{\zeta}^-)' \quad \beta_0^+ \quad \beta_0^- \quad (\boldsymbol{\beta}^+)' \quad (\boldsymbol{\beta}^-)' \quad (\boldsymbol{\rho}^+)'\ )' \\
\mathbf{c} := & (\ \mathbf{1}' \quad \mathbf{0}' \quad \mathbf{1}' \quad 0 \quad 0 \quad \mathbf{0}' \quad \mathbf{0}' \quad \mathbf{0}'\ )' \\
\mathbf{a} := & (\ \mathbf{0}' \quad \mathbf{0}' \quad \mathbf{0}' \quad 0 \quad 0 \quad \mathbf{0}' \quad \mathbf{0}' \quad \mathbf{1}'\ )' \\
\mathbf{A} := & \begin{bmatrix} \mathbf{I} & \mathbf{0} & -\mathbf{I} & \mathbf{1} & -\mathbf{1} & \mathbf{X} & -\mathbf{X} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{I} & -\mathbf{G} \end{bmatrix} \\
\mathbf{b} := & \begin{pmatrix} \boldsymbol{Y} \\ \mathbf{0} \end{pmatrix},
\end{array}
$$

where $\mathbf{G} := \begin{pmatrix} \mathbf{1}_{p_1} & \mathbf{0}_{p_1} & \cdots & \mathbf{0}_{p_1} \\ \mathbf{0}_{p_2} & \mathbf{1}_{p_2} & \cdots & \mathbf{0}_{p_2} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0}_{p_J} & \mathbf{0}_{p_J} & \cdots & \mathbf{1}_{p_J} \end{pmatrix}$.

Similarly, the optimization problem for penalized grouped quantile regression can be written as the same LP as the median regression except for the change in the cost vector

$$\mathbf{c} := (\tau\mathbf{1}', \mathbf{0}', (1-\tau)\mathbf{1}', 0, 0, \mathbf{0}', \mathbf{0}', \mathbf{0}')'.$$

Taking another example, if the ℓ_1 norm penalty for functional component selection in Section 4.3 is replaced with the OSCAR penalty on the recalibration parameters $\boldsymbol{\theta}$, the optimization problem of the structured SVM in (18) becomes

$$
\begin{cases}
\min\limits_{\boldsymbol{\zeta} \in \mathcal{R}^{nk}, \boldsymbol{\theta} \in \mathcal{R}^d} & \sum_{j=1}^k (\mathbf{L}^j)'(\boldsymbol{\zeta}^j)_+ + \frac{\lambda}{2} \sum_{\nu=1}^d \theta_\nu \left(\sum_{j=1}^k (\boldsymbol{\beta}^j)' \mathcal{K}_\nu \boldsymbol{\beta}^j \right) \\
& + \lambda_\theta \sum\limits_{1 \le \nu \le \mu \le d} \max(\theta_\nu, \theta_\mu) \\
\text{s.t.} & \sum_{\nu=1}^d \theta_\nu \mathcal{K}_\nu \boldsymbol{\beta}^j - \boldsymbol{\zeta}^j = \boldsymbol{y}^j - \beta_0^j \mathbf{1} \text{ for } j = 1, \ldots, k. \\
& \boldsymbol{\theta} \ge \mathbf{0}.
\end{cases} \tag{22}
$$

Introduce slack variables $\eta := \eta^+ - \eta^-$ for all pairwise differences $(\theta_i - \theta_j)$ for $1 \le i < j \le d$. Let $\boldsymbol{e}_i$ be the d-vector with its ith element equal to 1 and 0 elsewhere. And let $\boldsymbol{\Delta}$ denote a $d(d-1)/2 \times d$ matrix whose row vectors are $(\boldsymbol{e}_i - \boldsymbol{e}_j)$ for $1 \le i < j \le d$ in the order of $(\theta_i - \theta_j)$ in η. Then using the same notation as in Section 4.3, the new θ-step for the SVM with the OSCAR penalty can be rephrased as a parametric LP with

$$
\begin{array}{rl}
\boldsymbol{z} := & (\ \boldsymbol{\theta}' \quad (\boldsymbol{\eta}^+)' \quad (\boldsymbol{\eta}^-)' \quad (\boldsymbol{\zeta}^+)' \quad (\boldsymbol{\zeta}^-)'\ )' \\
\mathbf{c} := & (\ \mathbf{g}' \quad \mathbf{0}' \quad \mathbf{0}' \quad \mathbf{L}' \quad \mathbf{0}'\ )' \\
\mathbf{a} := & (\ (d+1)\mathbf{1}' \quad \mathbf{1}' \quad \mathbf{1}' \quad \mathbf{0}' \quad \mathbf{0}'\ )' \\
\mathbf{A} := & \begin{bmatrix} \mathbf{X} & \mathbf{0} & \mathbf{0} & -\mathbf{I} & \mathbf{I} \\ \boldsymbol{\Delta} & -\mathbf{I} & \mathbf{I} & \mathbf{0} & \mathbf{0} \end{bmatrix} \\
\mathbf{b} := & \begin{pmatrix} \boldsymbol{Y} - \boldsymbol{\beta}_0 \otimes \mathbf{1}_n \\ \mathbf{0} \end{pmatrix}.
\end{array}
$$

More examples can be found in Yao (2008).

4.5 Computation

The LP problems for the examples in Sections 4.1 - 4.3 share a similar structure that can be exploited in computation. First of all, the $\mathbf{A}$ matrix has both $\mathbf{I}$ and $-\mathbf{I}$ as its sub-matrices, and the entries of the penalty coefficient vector $\mathbf{a}$ corresponding to $\mathbf{I}$ and $-\mathbf{I}$ in $\mathbf{A}$ are zero. Thus, the ranks of $\mathbf{A}$ and $\mathbf{A}_{\mathcal{I}_\mathbf{a}}$ are M, and the initial optimal solution exists and can be easily identified. Due to the special structure of $\mathbf{A}_{\mathcal{I}_\mathbf{a}}$, it is easy to find a basic index set $\mathcal{B}^* \subset \mathcal{I}_\mathbf{a}$ for the initial LP problem in (9), which gives a feasible solution. For instance, a feasible basic solution can be obtained by constructing a basic index set $\mathcal{B}^*$ such that for $b_j \geq 0$, we choose the jth index from those for $\mathbf{I}$, and otherwise from the indices for $-\mathbf{I}$. For the θ-step of structured SVM, $\mathcal{B}^*$ itself is the initial optimal basic index set, and it gives a trivial initial solution. For the l_1-norm SVM and l_1-norm quantile regression, the basic index set defined above is not optimal. However, the initial optimal basic index set can be obtained easily from $\mathcal{B}^*$. In general, the tableau-simplex algorithm in Section 3 can be used to find the optimal basic index set of a standard LP problem, taking any feasible basic index set $\mathcal{B}$ as a starting point. Necessary modification of the algorithm for standard LP problems is that the entry index $j^l \in \mathcal{N}$ is chosen from any j with $a_j = 0$ and $\breve{\mathbf{c}}^l_j < 0$ at Step 3. For $\mathcal{B}^*$, all but the indices j for β_0^+ and β_0^- satisfy $\breve{\mathbf{c}}^l_j \geq 0$. Therefore, one of the indices for β_0 will move into the basic index set first by the algorithm, and it may take some iterations to get the initial optimal index set for the two regularization problems.

A tableau contains all the information on the current LP solution and the terms necessary for the next update. To discuss the computational complexity of the tableau updating algorithm in Section 3.2.2, let $\mathbf{T}^l$ denote the tableau, an $(N+1) \times (M+2)$ matrix associated with the current optimal basic index set $\mathcal{B}^l$. For a compact statement of the updating formula, assume that the tableau is rearranged such that the pivot columns and the pivot rows precede the zeroth column and the cost row and the penalty row, respectively. For the entry index j^l and exit index i^l defined in the algorithm, $\mathbf{T}^l_{j^l}$ denotes its j^lth column vector, $\mathbf{T}^{l\prime}_{i^l_*}$ the i^l_*th row vector of $\mathbf{T}^l$, and $T_{i^l_* j^l}$ the $i^l_* j^l$th entry of $\mathbf{T}^l$. The proof of Theorem 11 in Appendix D implies the following updating formula:

$$\mathbf{T}^{l+1} = \mathbf{T}^l - \frac{1}{T_{i^l_* j^l}} \left(\mathbf{T}^l_{j^l} - \boldsymbol{e}_{i^l_*} \right) \mathbf{T}^{l\prime}_{i^l_*}. \tag{23}$$

Therefore, the computational complexity of the tableau updating is approximately $O(MN)$ for each iteration in general.

For the three examples, tableau update can be further streamlined. Exploiting the structure of $\mathbf{A}$ with paired columns and fixed elements in the tableau associated with $\mathcal{B}^l$, we can compress each tableau, retaining the information about the current tableau, and update the reduced tableau instead. We leave discussion of implementation details elsewhere, but mention that updating such a reduced tableau has the complexity of $O((N_g - M)M)$ for each iteration, where N_g is the reduced number of columns in $\mathbf{A}$ counting only one for each of the paired columns. As a result, when the tableau algorithm stops in J iterations, the complexity of both l_1-norm SVM and l_1-norm QR as a whole is $O((p+1)nJ)$ while that of the θ-step of structured SVM is roughly $O(dnkJ)$, where p is the number of variables, d is the number of kernel functions, and k is the number of classes.

5 A Closer Look at the l_1-Norm Support Vector Machine

Taking the l_1-norm SVM as a case in point, we describe the implications of the tableau-simplex algorithm for generating the solution path. Zhu et al. (2004) provide a specific path-finding algorithm for the l_1-norm SVM in the complexity-bounded formulation of (7) and give a careful treatment of this particular problem.

We discuss the correspondence and generality of the tableau-simplex algorithm in comparison with their algorithm.

5.1 Status Sets

For the SVM problem with the complexity bound s (i.e. $\|\boldsymbol{\beta}\|_1 \leq s$), let $\beta_0(s)$ and $\boldsymbol{\beta}(s) := (\beta_1(s), \cdots, \beta_p(s))$ be the optimal solution at s. Zhu et al. (2004) categorize the variables and cases that are involved in the regularized LP problem as follows:

- Active set: $\mathcal{A}(s) := \{j : \beta_j(s) \neq 0,\ j = 0, 1, \ldots, p\}$
- Elbow set: $\mathcal{E}(s) := \{i : y_i\{\beta_0(s) + \mathbf{x}_i\boldsymbol{\beta}(s)\} = 1,\ i = 1, \ldots, n\}$
- Left set: $\mathcal{L}(s) := \{i : y_i\{\beta_0(s) + \mathbf{x}_i\boldsymbol{\beta}(s)\} < 1,\ i = 1, \ldots, n\}$
- Right set: $\mathcal{R}(s) := \{i : y_i\{\beta_0(s) + \mathbf{x}_i\boldsymbol{\beta}(s)\} > 1,\ i = 1, \ldots, n\}$.

Now, consider the solution $\boldsymbol{z}(s)$ given by the tableau-simplex algorithm as defined in Section 4.2 and the equality constraints of $\mathbf{A}\boldsymbol{z}(s) = \boldsymbol{b}$, that is,

$$\mathbf{A}\boldsymbol{z}(s) := \beta_0(s)\boldsymbol{Y} + \text{diag}(\boldsymbol{Y})\mathbf{X}\boldsymbol{\beta}(s) + \boldsymbol{\zeta}(s) = \mathbf{1}.$$

It is easy to see that for any solution $\boldsymbol{z}(s)$, its non-zero elements must be one of the following types, and hence associated with $\mathcal{A}(s)$, $\mathcal{L}(s)$, and $\mathcal{R}(s)$:

- $\beta_j^+(s) > 0$ or $\beta_j^-(s) > 0$ (but not both) $\Rightarrow j \in \mathcal{A}(s)$;
- $\zeta_i^+(s) > 0$ and $\zeta_i^-(s) = 0\ \Rightarrow i \in \mathcal{L}(s)$;
- $\zeta_i^+(s) = 0$ and $\zeta_i^-(s) > 0\ \Rightarrow i \in \mathcal{R}(s)$.

On the other hand, if $\zeta_i^+(s) = 0$ and $\zeta_i^-(s) = 0$, then $i \in \mathcal{E}(s)$, the elbow set.

5.2 Assumption

Suppose that the lth joint solution at $s = s^l$ is non-degenerate. Then $z_j(s^l) > 0$ if and only if $j \in \mathcal{B}^l$. This gives

$$|\mathcal{A}(s^l)| + |\mathcal{L}(s^l)| + |\mathcal{R}(s^l)| = n.$$

Since $\mathcal{E}(s) \cup \mathcal{L}(s) \cup \mathcal{R}(s) = \{1, \ldots, n\}$ for any s, the relationship that $|\mathcal{A}(s^l)| = |\mathcal{E}(s^l)|$ must hold for all the joint solutions. In fact, the equality of the cardinality of the active set and the elbow set is stated as an assumption for uniqueness of the solution in the algorithm of Zhu et al. (2004). The implicit assumption of $\boldsymbol{z}^l_{\mathcal{B}^l} > \mathbf{0}$ at each joint implies $\boldsymbol{z}^{l+1} \neq \boldsymbol{z}^l$, the non-degeneracy assumption for the simplex algorithm. Thus the simplex algorithm is less restrictive. In practice, the assumption that joint solutions are non-degenerate may not hold, especially when important predictors are discrete or coded categorical variables such as gender. For instance, the initial solution of the l_1-norm SVM violates the assumption in most cases, requiring a separate treatment for finding the next joint solution after initialization. In general, there could be more than one degenerate joint solutions along the solution path. This would make the tableau-simplex algorithm appealing as it does not rely on any restrictive assumption.

5.3 Duality in Algorithm

To move from one joint solution to the next, the simplex algorithm finds the entry index j^l. For the l_1-norm SVM, each index is associated with either β_j or ζ_i. Under the non-degeneracy assumption, the variable associated with j^l must change from zero to non-zero after the joint ($s > s^l$). Therefore, only one of the following "events" as defined in Zhu et al. (2004) can happen immediately after a joint solution:

- $\beta_j(s^l) = 0$ becomes $\beta_j(s) \neq 0$, i.e., an inactive variable becomes active;
- $\zeta_i(s^l) = 0$ becomes $\zeta_i(s) \neq 0$, i.e., an element leaves the elbow set and joins either the left set or the right set.

In conjunction with the entry index, the simplex algorithm determines the exit index, which accompanies one of the reverse events.

The algorithm in Zhu et al. (2004), driven by the Karush-Kuhn-Tucker optimality conditions, seeks the event with the smallest "$\Delta loss/\Delta s$," in other words, the one that decreases the cost with the fastest rate. The simplex algorithm is consistent with this existing algorithm. As in (10), recall that the entry index j^l is chosen to minimize $(\check{\mathrm{c}}^l_j/\check{\mathrm{a}}^l_j)$ among $j \in \mathcal{N} \setminus \mathcal{B}^l$ with $\check{\mathrm{a}}^l_j > 0$. $\mathcal{N} \setminus \mathcal{B}^l$ contains those indices corresponding to $j \notin \mathcal{A}(s^l)$ or $i \in \mathcal{E}(s^l)$. Analogous to the optimal moving direction $\boldsymbol{d}^l$ in (11), define $\boldsymbol{v}^j = (v^j_1, \ldots, v^j_N)'$ such that

$$\boldsymbol{v}^j_{\mathcal{B}^l} = -\mathbf{A}^{-1}_{\mathcal{B}^l}\mathbf{A}_j, v^j_j = 1, \text{ and } v^j_i = 0 \text{ for } i \in \mathcal{N} \setminus (\mathcal{B}^l \cup \{j\}).$$

Then $\check{\mathrm{a}}^l_j := (\mathrm{a}_j - \mathbf{a}'_{\mathcal{B}^l}\mathbf{A}^{-1}_{\mathcal{B}^l}\mathbf{A}_j) = \mathbf{a}'\boldsymbol{v}^j \propto \Delta s_j$ and $\check{\mathrm{c}}^l_j := (\mathrm{c}_j - \mathbf{c}'_{\mathcal{B}^l}\mathbf{A}^{-1}_{\mathcal{B}^l}\mathbf{A}_j) = \mathbf{c}'\boldsymbol{v}^j \propto \Delta loss_j$. Thus, the index chosen by the simplex algorithm in (10) maximizes the rate of reduction in the cost, $\Delta loss/\Delta s$.

The existing l_1-norm SVM path algorithm needs to solve roughly p groups of $|\mathcal{E}|$-variate linear equation systems for each iteration. Its computational complexity can be $O(p|\mathcal{E}|^2 + p|\mathcal{L}|)$ if Sherman-Morrison updating formula is used. On the other hand, the computational complexity of the tableau-simplex algorithm is $O(pn)$ for each iteration as mentioned in Section 4. Therefore, the former could be faster if n/p is large; otherwise, the tableau-simplex algorithm is faster.

Most of the arguments in this section also apply for the comparison of the simplex algorithms with the extended solution path algorithm for the l_1-norm multi-class SVM by Wang and Shen (2006).

6 Numerical Results

We illustrate the use of the tableau-simplex algorithm for parametric LP in statistical applications with a simulated example and analysis of real data, and discuss model selection or variable selection problems therein.

6.1 Quantile Regression

Quantile regression has been discussed in Sections 4.1 and 4.4. A simulation study is presented here to illustrate the use of the computational algorithm for quantile regression with different penalties and for their comparisons.

In the simulation study, 10 dimensional covariates are generated from the standard normal distribution independently, that is, $\mathbf{X} := (X_1, \ldots, X_{10}) \sim N(\mathbf{0}, \mathbf{I})$. The response variable is defined by $Y = \beta_0 + \mathbf{x}\boldsymbol{\beta} + \epsilon$ for some fixed β_0 and $\boldsymbol{\beta}$, where $\epsilon \sim N(0, \sigma^2)$, and $\mathbf{x}$ and ϵ are assumed to be mutually independent. The theoretical τth conditional quantile function is then given by $m_\tau(\mathbf{x}) = \sigma\Phi^{-1}(\tau) + \beta_0 + \mathbf{x}\boldsymbol{\beta}$, where Φ is the

cdf of the standard normal distribution. Restricting to linear functions only, suppose that an estimated τth conditional quantile function is $f(\mathbf{x}) = \hat{\beta}_0 + \mathbf{x}\hat{\boldsymbol{\beta}}$. Under the check function as a loss criterion, we can verify that the theoretical risk of f is given as

$$\begin{aligned} R(f;\beta_0,\boldsymbol{\beta}) &:= \mathrm{E}\left\{\tau(Y-\hat{\beta}_0-\mathbf{X}\hat{\boldsymbol{\beta}})_+ + (1-\tau)(Y-\hat{\beta}_0-\mathbf{X}\hat{\boldsymbol{\beta}})_-\right\} \\ &= \left\{\tau - \Phi\left(\frac{\hat{\beta}_0-\beta_0}{\sqrt{\sigma^2+\|\boldsymbol{\beta}-\hat{\boldsymbol{\beta}}\|_2^2}}\right)\right\}(\beta_0-\hat{\beta}_0) \\ &\quad + \sqrt{\frac{\sigma^2+\|\boldsymbol{\beta}-\hat{\boldsymbol{\beta}}\|_2^2}{2\pi}}\exp\left\{-\frac{(\hat{\beta}_0-\beta_0)^2}{2(\sigma^2+\|\boldsymbol{\beta}-\hat{\boldsymbol{\beta}}\|_2^2)}\right\}. \end{aligned} \tag{24}$$

For each τ, the risk of the true quantile function $m_\tau(\mathbf{x})$ is $(\sigma/\sqrt{2\pi})\exp\{-\Phi^{-1}(\tau)^2/2\}$, which represents the minimal achievable risk. Note that the maximum of the minimal risks in this case occurs when $\tau = 0.5$ (i.e., for the median), and the true conditional median function is $m_{0.5}(\mathbf{x}) = \beta_0 + \mathbf{x}\boldsymbol{\beta}$ with the risk of $\sigma/\sqrt{2\pi}$.

Suppose that the variables in the linear model form three groups, $\{1,2,3\}$, $\{4,5,6,7\}$, and $\{8,9,10\}$ of sizes $p_1 = 3$, $p_2 = 4$, and $p_3 = 3$, respectively, and they are alternatively indexed by (11,12,13), (21,22,23,24), and (31,32,33). Then the linear model can be restated as

$$Y = \beta_0 + \sum_{j=1}^{J}\sum_{i=1}^{p_j} x_{ij}\beta_{ij} + \epsilon$$

with the number of groups, $J = 3$. We set $\beta_0 = 0$, $\boldsymbol{\beta} := (\boldsymbol{\beta}_1', \boldsymbol{\beta}_2', \boldsymbol{\beta}_3')'$ with $\boldsymbol{\beta}_1 = (2,3,2)'$, $\boldsymbol{\beta}_2 = (0,0,0,0)'$, and $\boldsymbol{\beta}_3 = (-3,2,-2)'$, and $\sigma^2 = 50$. For the setting, the signal-to-noise ratio defined as $\mathrm{Var}(\mathbf{X}\boldsymbol{\beta})/\sigma^2$ is 0.68, and the minimal risk in estimating the median regression function is $\sigma/\sqrt{2\pi} \approx 2.821$.

In the study, 100 pairs of $\mathbf{x}$ and y were generated independently from the model. Focusing on the case with $\tau = 0.5$, we applied median regression with the ℓ_1 norm penalty in Section 4.1 and grouped median regression with the F_∞ norm penalty and the OSCAR penalty in Section 4.4 to the simulated data.

Figure 1 shows typical solution paths of grouped median regression with the F_∞ norm penalty indexed by s. The estimated coefficients are plotted in the left panel, and their absolute values are plotted in the right panel. They illustrate the general characteristic of penalized grouped regression that the coefficients in each group form a stem in the beginning and then branch out later for a better fit to the data. From the figure, we can see that the variable group 1 (in red) and the group 3 (in blue) stand out at the early stage of the solution path as expected.

The risk associated with the solution at each point of the paths is theoretically available for this example, and thus the optimal value of the regularization parameter can be defined. However, in practice, λ (or s) needs to be chosen data-dependently, and this gives rise to an important class of model selection problems in general. For the feasibility of data-dependent choice of the regularization parameter, we carried out cross validation and made comparison with the theoretically optimal values. The dashed lines in Figure 1 indicate the optimal value of s chosen by 10-fold cross validation under the check loss with $\tau = 0.5$. The left panel in Figure 2 displays the path of 10-fold cross validated risk in black for median regression with the F_∞ norm penalty corresponding to the coefficient paths in Figure 1 and the theoretical risk path in (24) in blue. The figure also shows the cross validated risk paths of the median regression fits with ℓ_1 norm and OSCAR penalties for comparison. In this case, the median regression fit with the F_∞ norm penalty produced the smallest cross validated risk. Note that we normalized the penalty parameter s in the figure for each of the

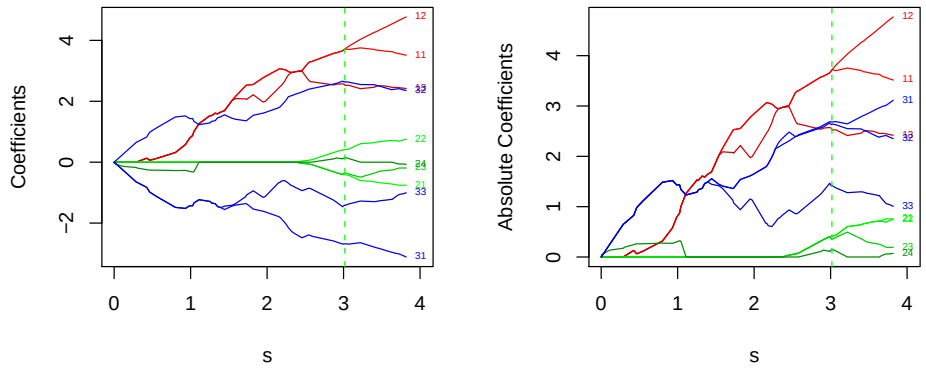

Figure 1: The solution paths of grouped median regression with the F_∞ norm penalty for simulated data. Colors (blue, green, and red) distinguish the three groups of variables. The left panel shows the regression coefficients while the right panel shows their absolute values. The vertical dashed line in each panel specifies the value s with the minimum of 10-fold cross validated risk under the check loss for $\tau = 0.5$.

three penalties so that the values of s are comparable across different penalties. Normalization was done by considering inherent difference in the expected size of each penalty for a given model. Specifically, the normalizing constants were determined such that the expected size of each penalty should be the same if β_j's are independent and identically distributed with a uniform distribution on $(-a/2, a/2)$ for any given $a > 0$.

To increase the smoothness of a risk path and the stability in identification of the optimal value of a tuning parameter in general, one may smooth out an individual cross validated risk curve or take the average of multiple curves over different splits of the data. To that effect, cross validation was repeated 20 times for averaging in our experiment, and Figure 2 shows, in fact, the average cross validated risk paths.

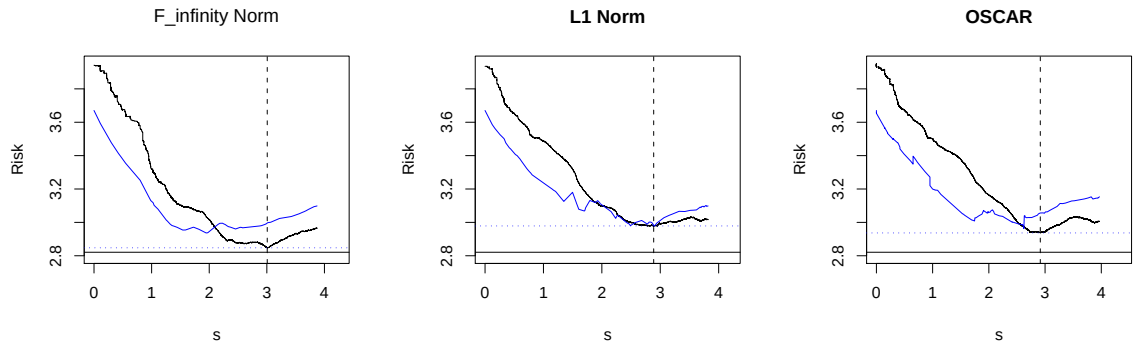

Figure 2: The estimated risk paths of median regression with F_∞ norm (left), ℓ_1 norm (center), and OSCAR penalty (right) by 10-fold cross validation for the simulated data used in Figure 1. In each panel, the black curve is the cross validated risk, the blue curve is the theoretical risk, and the vertical dashed line indicates the value s with the minimum cross validated risk. Horizontally, the dotted line corresponds to the minimum estimated risk, and the solid line marks the theoretically achievable minimum risk.

Parameter	True value	Penalty		
		F_∞ norm	ℓ_1 norm	OSCAR
β_{11}	2	1.8096	1.5990	1.4902
β_{12}	3	2.2862	2.5443	2.3869
β_{13}	2	1.6697	1.4905	1.4504
β_{21}	0	-0.0086	-0.0355	-0.0065
β_{22}	0	0.0015	-0.0058	0.0270
β_{23}	0	-0.0255	-0.0222	-0.0050
β_{24}	0	0.0163	0.0391	0.1200
β_{31}	-3	-2.3156	-2.5095	-2.3715
β_{32}	2	1.7857	1.5940	1.5525
β_{33}	-2	-1.7348	-1.5360	-1.4397

Table 1: The mean estimates of regression coefficients for the variables in median regression fits with F_∞ norm, ℓ_1 norm, and OSCAR penalties over 400 replicates of simulated data.

In order to compare the effect of the three different penalties on the accuracy of fitted median regression function, we generated 400 replicates of simulated data and repeated model fitting and selection by 10-fold cross validation. For each replicate, we chose the value s with minimum cross validated risk to identify the optimal model along the coefficient path of each method. Across the 400 replicates, median regression with the F_∞ norm penalty gave the mean minimum cross validated risk of 2.9551 with standard error of 0.0127 while the ℓ_1 norm penalty resulted in the mean risk of 3.0022 with standard error of 0.0133, and the OSCAR penalty had the mean risk of 3.0275 with standard error of 0.0131. The mean risk of the grouped median regression with F_∞ norm penalty is significantly smaller than those with ℓ_1 norm and OSCAR penalties in this example, probably due to the fact that the F_∞ group penalty directly utilizes the sparse structure of true regression coefficients.

However, the use of groupwise ℓ_∞ norm in the F_∞ penalty has an impact on the relative size of estimated coefficients. Table 1 shows the estimated regression coefficients for the 10 variables in median regression fits with F_∞ norm, ℓ_1 norm, and OSCAR penalties, respectively, averaged over 400 replicates. Compared to the individual ℓ_1 norm penalty, we can see that the F_∞ norm penalty tends to attenuate more extreme coefficients and produce values pulled toward the mean (in the absolute value) within each group. See, in particular, the variable groups 1 and 3 for the attenuation effect.

In addition, Table 2 summarizes the proportion of inclusion of each of the 10 variables in the median regression models fitted to those 400 replicates when the best model is chosen by 10-fold cross validation. Across the three different penalties, we see that the sensitivity of selecting a variable when it is active in the true median regression function is very high. However, the specificity of excluding a variable when it is indeed inactive is quite low. The F_∞ norm penalty gives the lowest specificity rates while it yields the highest sensitivity rates among the three penalties. As illustrated in Figure 1, a possible explanation is that cross validation tends to select models with extra variables in an attempt to improve prediction accuracy by allowing larger coefficients for the relevant variables.

6.2 Income Data Analysis

For a real application, we take the income data in Hastie et al. (2001), which are extracted from a marketing database for a survey conducted in the Bay area (1987). The data set is available at http://www-

Parameter	True value	Penalty		
		F_∞ norm	ℓ_1 norm	OSCAR
β_{11}	2	0.9925	0.9375	0.9025
β_{12}	3	1.0000	0.9925	0.9750
β_{13}	2	0.9875	0.9325	0.9050
β_{21}	0	0.8300	0.6150	0.6300
β_{22}	0	0.8350	0.6150	0.6225
β_{23}	0	0.8300	0.6050	0.6500
β_{24}	0	0.8125	0.6350	0.6375
β_{31}	-3	1.0000	0.9875	0.9725
β_{32}	2	1.0000	0.9300	0.9250
β_{33}	-2	0.9900	0.9175	0.9125

Table 2: The inclusion proportions of variables in the fitted median regression models with F_∞ norm, ℓ_1 norm, and OSCAR penalties for 400 replicates of simulated data.

stat.stanford.edu/∼tibs/ElemStatLearn/. It consists of 14 demographic attributes with a mixture of categorical and continuous variables, which include age, gender, education, occupation, marital status, householder status (own home/rent/other), and annual income among others. The main goal of the analysis is to predict the annual income of the household (or personal income if single) from the other 13 demographics attributes.

The original response of the annual income takes one of the following income brackets: < 10, [10, 15), [15, 20), [20, 25), [25, 30), [30, 40), [40, 50), [50, 75), and ≥ 75 in the unit of \$1,000. For simplification, we created a proxy numerical response by converting each bracket into its middle value except the first and the last ones, which were mapped to some reasonable values albeit arbitrary. Removing the records with missing values yields a total of 6,876 records. Because of the granularity in the response, the normal-theory regression would not be appropriate. As an alternative, we consider median regression, in particular, l_1 norm median regression and grouped median regression for simultaneous variable selection and prediction. In the analysis, each categorical variable with k categories was coded by (k-1) 0-1 dummy variables with the majority category treated as the baseline. Some genuinely numerical but bracketed predictors such as age were also coded similarly as the response. As a result, 35 variables were generated from the 13 original variables.

The data set was split into a training set of 2,000 observations and a test set of 4,876 for evaluation. All the predictors were centered to zero and scaled to have the squared norm equal to the training sample size before fitting a model. Inspection of the marginal associations of the original attributes with the response necessitated inclusion of a quadratic term for age. We then considered linear median regression with the main effect terms only (35 variables plus the quadratic term) and with additional two-way interaction terms. There are potentially 531 two-way interaction terms by taking the product of each pair of the normalized main effect terms from different attributes. In an attempt to exclude nearly constant terms, we screened out any product with the relative frequency of its mode 90% or above. This resulted in addition of 69 two-way interactions to the main effects model. Note that the interaction terms were put in the partial two-way interaction model without further centering and normalization for the clarity of the model. Approximately three quarters of the interactions had their norms within 10% difference from that of the main effects.

Figure 3 shows the coefficient paths of the main effects model with ℓ_1 penalty in the left panel and that with F_∞ group penalty for the training data set. The coefficients of the dummy variables grouped

for each categorical variable are of the same color. In both models, several variables emerge at the early stage as important predictors of the household income and remain important throughout the paths. Note the visible effect of the F_∞ group penalty on the coefficients of homeownership (hs.own and hs.withFamily) for small values of s in contrast with ℓ_1 penalty. Among those, the factors positively associated with household income are home ownership (relative to renting), education, dual income due to marriage (relative to 'not married'), age, and being male. Marital status and occupation are also strong predictors. As opposed to those positive factors, being single or divorced (relative to 'married') and being a student, clerical worker, retired or unemployed (relative to professionals/managers) are negatively associated with the income. So is the quadratic term of age as expected. In general, it would be too simplistic to assume that the demographic factors in the data affect the household income additively. Truthful models would need to take into account some high order interactions, reflecting the socio-economic fabric of the household income structure. Some of the two-way interactions worthwhile to mention are 'dual income $*$ home ownership', 'home ownership $*$ education', and 'married but no dual income $*$ education' with positive coefficients, and 'single $*$ education' and 'home ownership $*$ age' with negative coefficients.

As in the quantile regression simulation, we chose optimal values of s by cross validation with the absolute deviation loss. Five-fold cross validation was repeated 5 times for different splits of the training data, and the resulting risks were averaged. Figure 4 displays the paths of actual risks over the test set for the main effect models (left) and for the partial two-way interaction models (right) using ℓ_1 norm median regression. The dashed lines indicate the minimizers s of the averaged risks and the solid lines those of the actual risks over the test set. Cross validation seems to give a reasonable choice of s in terms of risk. Note that there is a range of optimal values with about the same risk in both panels, which suggests that one may as well average the models in the range. A notable difference between the risk paths is the amount of regularization desired to attain the minimum risk in comparison with the full models. That is, regularization improves the two-way interaction models much more than the main effects models. Moreover, the selected two-way interaction model has a smaller risk over the test set than the main effect model in accordance with our understanding of the data. On the basis of evaluation over the test data, 95% confidence intervals of the true risk associated with the main effects and the two-way interaction models selected by the CV criteria are 7.799 ± 0.238 and 7.653 ± 0.236, respectively. In particular, a 95% confidence interval of the risk difference of the main effects model from the two-way model is given by 0.146 ± 0.0585, which indicates that the latter improves the former significantly in terms of the risk. We carried out similar analysis with F_∞ group penalty for the main effect model and the partial two-way interaction model, and observed reduction in the risk by the two-way interaction model. However, in comparison with the plain ℓ_1 norm penalty, the group penalty did not bring any particular advantage in reducing the risk and providing a better model. With F_∞ norm penalty, 95% confidence intervals of the true risk are 7.813 ± 0.241 for the main effect model and 7.780 ± 0.238 for the two-way interaction model, respectively.

7 Discussion

Tapping into a rich theory of linear programming and its algorithmic developments, we have provided a broad and unified perspective on the properties of solutions to a wide family of regularization methods for feature selection. We have shown that the solutions can be characterized completely by using the parametric linear programming techniques.

As for computational implementation, a single umbrella procedure can serve for all of the methods in the family in order to generate the entire set of regularized solutions. Capitalizing on the generality of our formulation, we have implemented the path generating algorithms presented in Section 4 with a modular

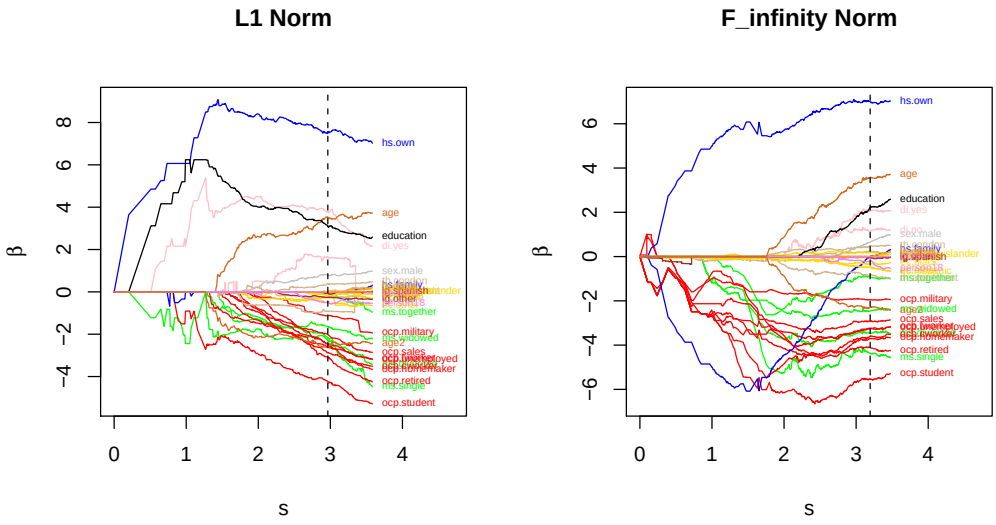

Figure 3: The coefficient paths of the main effects model with ℓ_1 penalty (left) and that with F_∞ group penalty (right) for the income data. For each categorical variable, the coefficients of the corresponding dummy variables are plotted as a group of the same color. The dashed lines indicate the models chosen by five-fold cross validation with the absolute deviation loss.

treatment relying on a core algorithm for the tableau-simplex method. `lpRegPath` is an R package for the implementation and currently available at http://www.stat.osu.edu/∼yao/software.html. Other extensions can be easily added to the package by using the core algorithm. Efficiency can be gained further when the umbrella procedure is tailored to each individual method by utilizing the structure of the computational elements specific to the method. Handling large scale data with a regularization method is a computational challenge in itself. For example, Kim et al. (2007) and Koh et al. (2007) discuss solving large scale ℓ_1 regularized least squares problem and logistic regression problem with interior point method. Making path generating algorithms scalable with the sample size and the dimension of features would be another direction to pursue.

As illustrated, the solution paths offer rich information about how constrained models evolve with features. Especially, they make it easy to recognize persistent features in the data, which are of general interest in data analysis. In addition to facilitating computation and tuning, the path-finding algorithms for feature selection can equip the data analyst with a useful tool for visualization of a model path. Combined with risk measures, such a path can portray a full spectrum of potentially good models for selection and averaging.

This paper has focused on elucidating the link between computational problems with linear constraints for feature selection in statistics and the linear programming theory. Beside those examples discussed in this paper, there remain many possible applications of the parametric linear programming techniques. For example, feature selection in nonparametric settings is worthwhile to investigate separately. Such an algorithm that explores a vast model space by gradually elaborating and selecting functional components or the kernels generating them will be a valuable extension of its parametric counterpart for modeling and prediction.

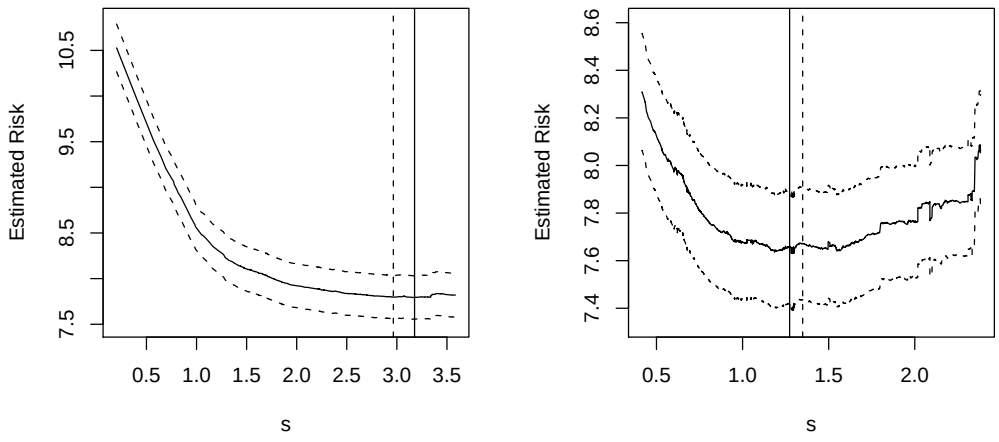

Figure 4: The paths of actual risks estimated over the test set and their 95% confidence intervals for the main effect models (left) and the partial two-way interaction models (right) with ℓ_1 norm median regression. The solid vertical lines mark the minimum values. The dashed vertical lines indicate the values of s minimizing the average risks from five-fold cross validation repeated five times.

Appendix

A Lemma 13

Lemma 13 *Suppose that* $\mathcal{B}^{l+1} := \mathcal{B}^l \cup \{j^l\} \setminus \{i^l\}$, *where* $i^l := B^l_{i^l_*}$. *Let* $\boldsymbol{d}^l$ *be defined as in* (11). *Then*

$$\boldsymbol{z}^{l+1} = \boldsymbol{z}^l - \frac{z^l_{i^l}}{d^l_{i^l}} \boldsymbol{d}^l.$$

Proof First observe that

$$\boldsymbol{z}^{l+1}_{\mathcal{B}^{l+1}} = \mathbf{A}^{-1}_{\mathcal{B}^{l+1}} \mathbf{b} = \mathbf{A}^{-1}_{\mathcal{B}^{l+1}} \mathbf{A}_{\mathcal{B}^l} \mathbf{A}^{-1}_{\mathcal{B}^l} \mathbf{b} = [\mathbf{A}^{-1}_{\mathcal{B}^{l+1}} \mathbf{A}_{\mathcal{B}^l}] \boldsymbol{z}^l_{\mathcal{B}^l}.$$

Without loss of generality, the i^l_*th column vector $\mathbf{A}_{i^l}$ of $\mathbf{A}_{\mathcal{B}^l}$ is replaced with $\mathbf{A}_{j^l}$ to give $\mathbf{A}_{\mathcal{B}^{l+1}}$. For the $\mathbf{A}_{\mathcal{B}^{l+1}}$,

$$\begin{aligned} [\mathbf{A}^{-1}_{\mathcal{B}^{l+1}} \mathbf{A}_{\mathcal{B}^l}]^{-1} &= \mathbf{A}^{-1}_{\mathcal{B}^l} \mathbf{A}_{\mathcal{B}^{l+1}} \\ &= [\mathbf{e}_1, \cdots, \mathbf{e}_{i^l_*-1}, \boldsymbol{u}^l, \mathbf{e}_{i^l_*+1}, \cdots, \mathbf{e}_M] \\ &= \begin{bmatrix} 1 & & u^l_1 & & \\ & \ddots & \vdots & & \\ & & u^l_{i^l_*} & & \\ & & \vdots & \ddots & \\ & & u^l_M & & 1 \end{bmatrix}, \end{aligned} \tag{25}$$

where $\boldsymbol{u}^l := \mathbf{A}_{\mathcal{B}^l}^{-1}\mathbf{A}_{j^l} = -\boldsymbol{d}^l_{\mathcal{B}^l}$. Thus, we have

$$\mathbf{A}_{\mathcal{B}^{l+1}}^{-1}\mathbf{A}_{\mathcal{B}^l} = \begin{bmatrix} 1 & & -\frac{u^l_1}{u^l_{i^l_*}} & & \\ & \ddots & \vdots & & \\ & & \frac{1}{u^l_{i^l_*}} & & \\ & & \vdots & \ddots & \\ & & -\frac{u^l_M}{u^l_{i^l_*}} & & 1 \end{bmatrix}. \tag{26}$$

Then it immediately follows that

$$\boldsymbol{z}^{l+1}_{\mathcal{B}^{l+1}} = \boldsymbol{z}^l_{\mathcal{B}^l} - \frac{z^l_{i^l}}{d^l_{i^l}}\boldsymbol{d}^l_{\mathcal{B}^l} - \frac{z^l_{i^l}}{d^l_{i^l}}\boldsymbol{e}_{i^l_*}.$$

Hence, $\boldsymbol{z}^{l+1} = \boldsymbol{z}^l - (z^l_{i^l}/d^l_{i^l})\boldsymbol{d}^l$. ■

B Proof of (13)

For $l = 0, \cdots, J-1$, consider the following difference

$$\begin{aligned} & \left[(\mathbf{c} + \lambda_l\mathbf{a}) - \mathbf{A}'\left(\mathbf{A}_{\mathcal{B}^{l+1}}^{-1}\right)'(\mathbf{c}_{\mathcal{B}^{l+1}} + \lambda_l\mathbf{a}_{\mathcal{B}^{l+1}})\right] - \left[(\mathbf{c} + \lambda_l\mathbf{a}) - \mathbf{A}'\left(\mathbf{A}_{\mathcal{B}^l}^{-1}\right)'(\mathbf{c}_{\mathcal{B}^l} + \lambda_l\mathbf{a}_{\mathcal{B}^l})\right] \\ = \; & -\mathbf{A}'\left(\mathbf{A}_{\mathcal{B}^l}^{-1}\right)'\left(\mathbf{A}_{\mathcal{B}^{l+1}}^{-1}\mathbf{A}_{\mathcal{B}^l}\right)'(\mathbf{c}_{\mathcal{B}^{l+1}} + \lambda_l\mathbf{a}_{\mathcal{B}^{l+1}}) + \mathbf{A}'\left(\mathbf{A}_{\mathcal{B}^l}^{-1}\right)'(\mathbf{c}_{\mathcal{B}^l} + \lambda_l\mathbf{a}_{\mathcal{B}^l}). \end{aligned}$$

By the intermediate calculation in Lemma 13, we can show that the difference is $\kappa_{\lambda_l}\mathbf{A}'\left(\mathbf{A}_{\mathcal{B}^l}^{-1}\right)'\mathbf{e}_{i^l_*}$, where

$$\begin{aligned} \kappa_{\lambda_l} \; &:= \; (\mathrm{c}_{B_{i^l_*}} + \lambda_l\mathrm{a}_{B_{i^l_*}}) - \frac{\mathrm{c}_{j^l} + \lambda_l\mathrm{a}_{j^l}}{u^l_{i^l_*}} + \sum_{i \in \left(\mathcal{B}^{l+1} \setminus \{j^l\}\right)} \frac{(\mathrm{c}_i + \lambda_l\mathrm{a}_i)u^l_i}{u^l_{i^l_*}} \\ &= \; \frac{(\mathbf{c}_{\mathcal{B}^l} + \lambda_l\mathbf{a}_{\mathcal{B}^l})'\mathbf{A}_{\mathcal{B}^l}^{-1}\mathbf{A}_{j^l} - (\mathrm{c}_{j^l} + \lambda_l\mathrm{a}_{j^l})}{u^l_{i^l_*}} \\ &= \; -\frac{\check{\mathrm{c}}^l_{j^l} + \lambda_l\check{\mathrm{a}}^l_{j^l}}{u^l_{i^l_*}}. \end{aligned}$$

Since $\lambda_l := -\check{\mathrm{c}}^l_{j^l}/\check{\mathrm{a}}^l_{j^l}$, $\kappa_{\lambda_l} = 0$, which proves (13).

C Proof of Theorem 7

Let $\mathfrak{B}^l := \mathcal{B}^l \cup \{j^l\}$ for $l = 0, \cdots, J-1$, and $\mathfrak{B}^J := \mathcal{B}^J \cup \{N+1\}$, where $\mathcal{B}^l$, $\mathcal{B}^J$, and j^l are as defined in the simplex algorithm. We will show that, for any fixed $s \in [s_l, s_{l+1})$ (or $s \geq s_J$), $\mathfrak{B}^l$ (or $\mathfrak{B}^J$) is an optimal basic index set for the LP problem in (8).

For simplicity, let $j^J := N+1$, $\mathrm{c}_{N+1} := 0$, $\mathbf{A}_{N+1} := \mathbf{0}$, and $\mathrm{a}_{N+1} := 1$. The inverse of

$$\mathbb{A}_{\mathfrak{B}^l} = \begin{bmatrix} \mathbf{A}_{\mathcal{B}^l} & \mathbf{A}_{j^l} \\ \mathbf{a}_{\mathcal{B}^l}{}' & \mathrm{a}_{j^l} \end{bmatrix}$$

is given by

$$\mathbb{A}_{\mathfrak{B}^l}^{-1} = \begin{bmatrix} \mathbf{A}_{\mathcal{B}^l}^{-1} & \mathbf{0} \\ \mathbf{0}' & 0 \end{bmatrix} + \frac{1}{\mathrm{a}_{j^l} - \mathbf{a}_{\mathcal{B}^l}{}' \mathbf{A}_{\mathcal{B}^l}^{-1} \mathbf{A}_{j^l}} \begin{bmatrix} -\mathbf{A}_{\mathcal{B}^l}^{-1} \mathbf{A}_{j^l} \\ 1 \end{bmatrix} \begin{bmatrix} -\mathbf{a}_{\mathcal{B}^l}{}' \mathbf{A}_{\mathcal{B}^l}^{-1} \\ 1 \end{bmatrix}'$$

for $l = 0, \cdots, J$.

First, we show that $\mathbb{A}_{\mathfrak{B}^l}$ is a feasible basic index set of (8) for $s \in [s_l, s_{l+1}]$, i.e.

$$\mathbb{A}_{\mathfrak{B}^l}^{-1}(\mathbb{b} + s\mathbb{b}^{*\prime}) \geq \mathbf{0}. \tag{27}$$

Recalling that $\boldsymbol{z}_{\mathcal{B}^l}^l = \mathbf{A}_{\mathcal{B}^l}^{-1}\mathbf{b}$, $z_{j^l}^l = 0$, $s_l = \mathbf{a}'\boldsymbol{z}^l = (\mathbf{a}_{\mathcal{B}}^l{}' \mathbf{A}_{\mathcal{B}^l}^{-1}\mathbf{b})$, $\boldsymbol{d}_{\mathcal{B}^l}^l = -\mathbf{A}_{\mathcal{B}^l}^{-1}\mathbf{A}_{j^l}$, and $d_{j^l}^l = 1$, we have

$$\begin{aligned} \mathbb{A}_{\mathfrak{B}^l}^{-1}(\mathbb{b} + s\mathbb{b}^{*\prime}) &= \mathbb{A}_{\mathfrak{B}^l}^{-1} \left\{ \begin{bmatrix} \mathbf{b} \\ 0 \end{bmatrix} + s \begin{bmatrix} \mathbf{0} \\ 1 \end{bmatrix} \right\} \\ &= \begin{bmatrix} \mathbf{A}_{\mathcal{B}^l}^{-1}\mathbf{b} \\ 0 \end{bmatrix} + \frac{(s - \mathbf{a}_{\mathcal{B}^l}{}' \mathbf{A}_{\mathcal{B}^l}^{-1}\mathbf{b})}{\mathrm{a}_{j^l} - \mathbf{a}_{\mathcal{B}^l}{}' \mathbf{A}_{\mathcal{B}^l}^{-1} \mathbf{A}_{j^l}} \begin{bmatrix} -\mathbf{A}_{\mathcal{B}^l}^{-1}\mathbf{A}_{j^l} \\ 1 \end{bmatrix} \\ &= \begin{bmatrix} \boldsymbol{z}_{\mathcal{B}^l}^l \\ z_{j^l}^l \end{bmatrix} + \frac{s - s_l}{\mathrm{a}_{j^l} + \mathbf{a}_{\mathcal{B}^l}{}' \boldsymbol{d}_{\mathcal{B}^l}^l} \begin{bmatrix} \boldsymbol{d}_{\mathcal{B}^l}^l \\ d_{j^l}^l \end{bmatrix}. \end{aligned} \tag{28}$$

From $\boldsymbol{z}^{l+1} - \boldsymbol{z}^l = -(z_{i^l}^l / d_{i^l}^l)\boldsymbol{d}^l$ and $s_{l+1} - s_l = \mathbf{a}'(\boldsymbol{z}^{l+1} - \boldsymbol{z}^l) = -(z_{i^l}^l / d_{i^l}^l)(\mathrm{a}_{j^l} + \mathbf{a}_{\mathcal{B}^l}{}' \boldsymbol{d}_{\mathcal{B}^l}^l)$, it can be shown that

$$(28) = \begin{bmatrix} \boldsymbol{z}_{\mathcal{B}^l}^l \\ z_{j^l}^l \end{bmatrix} + \frac{s - s_l}{s_{l+1} - s_l} \left\{ \begin{bmatrix} \boldsymbol{z}_{\mathcal{B}^l}^{l+1} \\ z_{j^l}^{l+1} \end{bmatrix} - \begin{bmatrix} \boldsymbol{z}_{\mathcal{B}^l}^l \\ z_{j^l}^l \end{bmatrix} \right\}.$$

Thus, (28) is a convex combination of $\boldsymbol{z}^l$ and $\boldsymbol{z}^{l+1}$ for $s \in [s_l, s_{l+1}]$, and hence it is non-negative. This proves the feasibility of $\mathbb{A}_{\mathfrak{B}^l}$ for $s \in [s_l, s_{l+1}]$ and $l = 0, \ldots, J-1$. For $s \geq s^J$, we have

$$\begin{aligned} & \mathbb{A}_{\mathfrak{B}^J}^{-1} \left\{ \begin{bmatrix} \mathbf{b} \\ 0 \end{bmatrix} + s \begin{bmatrix} \mathbf{0} \\ 1 \end{bmatrix} \right\} \\ = & \begin{bmatrix} \mathbf{A}_{\mathcal{B}^J}^{-1}\mathbf{b} \\ 0 \end{bmatrix} + (s - \mathbf{a}_{\mathcal{B}^J}{}' \mathbf{A}_{\mathcal{B}^J}^{-1}\mathbf{b}) \begin{bmatrix} \mathbf{0} \\ 1 \end{bmatrix} \\ = & \begin{bmatrix} \mathbf{A}_{\mathcal{B}^J}^{-1}\mathbf{b} \\ 0 \end{bmatrix} + (s - s_J) \begin{bmatrix} \mathbf{0} \\ 1 \end{bmatrix} \geq \mathbf{0}. \end{aligned}$$

Next, we prove that $\mathbb{A}_{\mathfrak{B}^l}$ is an optimal basic index set of (8) for $s \in [s_l, s_{l+1}]$ by showing $\mathbb{c} - \mathbb{A}'(\mathbb{A}_{\mathfrak{B}^l}^{-1})'\mathbb{c}_{\mathfrak{B}^l} \geq \mathbf{0}$. For $i = 1, \cdots, N$, the ith element of $\mathbb{c} - \mathbb{A}'(\mathbb{A}_{\mathfrak{B}^l}^{-1})'\mathbb{c}_{\mathfrak{B}^l}$ is

$$\begin{aligned} & \mathrm{c}_i - \begin{bmatrix} \mathbf{c}_{\mathcal{B}^l} \\ \mathrm{c}_{j^l} \end{bmatrix}' \mathbb{A}_{\mathfrak{B}^l}^{-1} \begin{bmatrix} \mathbf{A}_i \\ \mathrm{a}_i \end{bmatrix} \\ = & \mathrm{c}_i - \mathbf{c}_{\mathcal{B}^l}{}' \mathbf{A}_{\mathcal{B}^l}^{-1}\mathbf{A}_i - \frac{\mathrm{c}_{j^l} - \mathbf{c}_{\mathcal{B}^l}{}' \mathbf{A}_{\mathcal{B}^l}^{-1}\mathbf{A}_{j^l}}{\mathrm{a}_{j^l} - \mathbf{a}_{\mathcal{B}^l}{}' \mathbf{A}_{\mathcal{B}^l}^{-1}\mathbf{A}_{j^l}} (\mathrm{a}_i - \mathbf{a}_{\mathcal{B}^l}{}' \mathbf{A}_{\mathcal{B}^l}^{-1}\mathbf{A}_i) \\ = & \begin{cases} \check{\mathrm{c}}_i^l + \lambda_l \check{\mathrm{a}}_i^l & \text{for } i = 1, \cdots, N \\ \lambda_l & \text{for } i = N+1. \end{cases} \end{aligned}$$

Similarly, for $s \geq s^J$,

$$\begin{aligned} \mathrm{c}_i - \begin{bmatrix} \mathbf{c}_{\mathcal{B}^J} \\ 0 \end{bmatrix}' \mathbb{A}_{\mathfrak{B}^J}^{-1} \begin{bmatrix} \mathbf{A}_i \\ \mathrm{a}_i \end{bmatrix} &= \mathrm{c}_i - \mathbf{c}_{\mathcal{B}^J}{}' \mathbf{A}_{\mathcal{B}^J}^{-1} \mathbf{A}_i \\ &= \begin{cases} \check{\mathrm{c}}_i^J & \text{for } i = 1, \cdots, N \\ 0 & \text{for } i = N+1. \end{cases} \end{aligned}$$

Clearly, the optimality condition holds by the non-negativity of all the elements as defined in the simplex algorithm. This completes the proof.

D Proof of Theorem 11

i) By (26), we can update the pivot rows of the tableau as follows:

$$\begin{aligned} &(\text{the } i\text{th pivot row of } \mathcal{B}^{l+1}) \\ = &\begin{cases} (\text{the } i\text{th pivot row of } \mathcal{B}^l) - \frac{u_i^l}{u_{i_*^l}^l} (\text{the } i_*^l\text{th pivot row of } \mathcal{B}^l) & \text{for } i \neq i_*^l; \\ \frac{1}{u_{i_*^l}^l} (\text{the } i_*^l\text{th pivot row of } \mathcal{B}^l) & \text{for } i = i_*^l. \end{cases} \end{aligned} \tag{29}$$

If $u_i^l = 0$, the ith pivot row of $\mathcal{B}^{l+1}$ is the same as the ith pivot row of $\mathcal{B}^l (\overset{L}{>} \mathbf{0})$. For $i = i_*^l$, the ith pivot row of $\mathcal{B}^{l+1}$ is $(1/u_{i_*^l}^l)$ (the ith pivot row of $\mathcal{B}^l$) $\overset{L}{>} \mathbf{0}$. If $i \neq i_*^l$ and $u_i^l < 0$, which imply $-u_i^l/u_{i_*^l}^l > 0$, the ith pivot row of $\mathcal{B}^{l+1} \overset{L}{>} \mathbf{0}$ since the sum of any two lexicographically positive vectors is still lexicographically positive. According to the tableau update algorithm, we have $u_{i_*^l}^l > 0$, where i_*^l is the index number of the lexicographically smallest pivot row among all the pivot rows for $\mathcal{B}^l$ with $u_i^l > 0$. For $i \neq i_*^l$ and $u_i^l > 0$, by the definition of i_*^l,

$$\frac{\text{the } i_*^l\text{th pivot row of } \mathcal{B}^l}{u_{i_*^l}^l} \overset{L}{<} \frac{\text{the } i\text{th pivot row of } \mathcal{B}^l}{u_i^l}.$$

This implies that

$$\begin{aligned} &(\text{the } i\text{th pivot row for } \mathcal{B}^{l+1}) \\ = &\left(\text{the } i\text{th pivot row of } \mathcal{B}^l\right) - \frac{u_i^l}{u_{i_*^l}^l} \left(\text{the } i_*^l\text{th pivot row of } \mathcal{B}^l\right) \overset{L}{>} \mathbf{0}. \end{aligned}$$

Therefore, all the updated pivot rows are lexicographically positive.

Remark 14 *If $z_{i^l}^l = 0$, (29) implies that $z_{B_i^l}^l = z_{B_i^l}^{l+1}$ for $i \neq i_*^l$, $i \in \mathcal{M}$. and $z_{j^l}^{l+1} = 0$. Hence $\boldsymbol{z}^{l+1} = \boldsymbol{z}^l$. On the other hand, if $z_{i^l}^l > 0$, $z_{j^l}^{l+1} = (z_{i^l}^l / u_{j^l}^l) > 0$ while $z_{j^l}^l = 0$ since $j^l \notin \mathcal{B}^l$. This implies $\boldsymbol{z}^{l+1} \neq \boldsymbol{z}^l$. Therefore, $\boldsymbol{z}^{l+1} = \boldsymbol{z}^l$ if and only if $z_{i^l}^l = 0$.*

ii) When the basic index set $\mathcal{B}^l$ is updated to $\mathcal{B}^{l+1}$, $\check{\mathrm{c}}_{j^l}^l < 0$. Since $j^l \in \mathcal{B}^{l+1}$, $\check{\mathrm{c}}_{j^l}^{l+1} = 0$ Then, $(\mathbf{c}_{j^l} - \mathbf{c}'_{\mathcal{B}^{l+1}} \mathbf{A}_{\mathcal{B}^{l+1}}^{-1} \mathbf{A}_{j^l}) - (\mathbf{c}_{j^l} - \mathbf{c}'_{\mathcal{B}^l} \mathbf{A}_{\mathcal{B}^l}^{-1} \mathbf{A}_{j^l}) = (\check{\mathrm{c}}_{j^l}^{l+1} - \check{\mathrm{c}}_{j^l}^l) > 0$.

Similarly as the proof of (13),

$$\left(\mathbf{c}' - \mathbf{c}'_{\mathcal{B}^{l+1}}\mathbf{A}^{-1}_{\mathcal{B}^{l+1}}\mathbf{A}\right) - \left(\mathbf{c}' - \mathbf{c}'_{\mathcal{B}^{l}}\mathbf{A}^{-1}_{\mathcal{B}^{l}}\mathbf{A}\right) = \kappa^l \mathbf{e}'_{i^l_*}\mathbf{A}^{-1}_{\mathcal{B}^{l}}\mathbf{A},$$

where $\kappa^l := (\mathbf{c}'_{\mathcal{B}^l}\boldsymbol{u}^l - \mathrm{c}_{j^l})/u^l_{i^l_*}$. $\mathbf{e}'_{i^l_*}\mathbf{A}^{-1}_{\mathcal{B}^l}\mathbf{A}$ is the i^l_*th pivot row for $\mathcal{B}^l$, which is lexicographically positive. Since the j^lth entry of $\mathbf{e}'_{i^l_*}\mathbf{A}^{-1}_{\mathcal{B}^l}\mathbf{A}$ is strictly positive, that of $(\mathbf{c}' - \mathbf{c}'_{\mathcal{B}^{l+1}}\mathbf{A}^{-1}_{\mathcal{B}^{l+1}}\mathbf{A}) - (\mathbf{c}' - \mathbf{c}'_{\mathcal{B}^l}\mathbf{A}^{-1}_{\mathcal{B}^l}\mathbf{A})$ must share the same sign with κ^l. Thus, we have $\kappa^l > 0$. Then the updated cost row is given as

$$\begin{aligned} & \left[-\mathbf{c}'_{\mathcal{B}^{l+1}}\mathbf{A}^{-1}_{\mathcal{B}^{l+1}}\mathbf{b}, \mathbf{c}' - \mathbf{c}'_{\mathcal{B}^{l+1}}\mathbf{A}^{-1}_{\mathcal{B}^{l+1}}\mathbf{A}\right] \\ = & \left[-\mathbf{c}'_{\mathcal{B}^{l}}\mathbf{A}^{-1}_{\mathcal{B}^{l}}\mathbf{b}, \mathbf{c}' - \mathbf{c}'_{\mathcal{B}^{l}}\mathbf{A}^{-1}_{\mathcal{B}^{l}}\mathbf{A}\right] + \kappa^l \mathbf{e}'_{i^l_*}\left[\mathbf{A}^{-1}_{\mathcal{B}^{l}}\mathbf{b}, \mathbf{A}^{-1}_{\mathcal{B}^{l}}\mathbf{A}\right]. \end{aligned}$$

Clearly, the cost row for $\mathcal{B}^{l+1}$ is lexicographically greater than that for $\mathcal{B}^l$.

References

Barrodale, I. and Roberts, F. (1973). An improved algorithm for discrete l_1 linear approximation, *SIAM Journal on Numerical Analysis* **10**: 839–848.

Bertsimas, D. and Tsitsiklis, J. (1997). *Introduction to Linear Programming*, Athena Scientific, Belmont, Massachusetts.

Bloomfield, P. and Steiger, W. (1980). Least absolute deviations curve-fitting, *SIAM Journal on Scientific Computing* **1**: 290–301.

Bloomfield, P. and Steiger, W. L. (1983). *Least Absolute Deviations: Theory, Applications, and Algorithms*, Birkhäuser.

Bondell, H. and Reich, B. (2008). Simultaneous regression shrinkage, variable selection, and supervised clustering of predictors with oscar, *Biometrics* **64**: 115–123.

Bradley, P. S. and Mangasarian, O. L. (1998). Feature selection via concave minimization and support vector machines, *in* J. Shavlik (ed.), *Machine Learning Proceedings of the Fifteenth International Conference*, Morgan Kaufmann, San Francisco, California, pp. 82–90.

Candes, E. and Tao, T. (2007). The Dantzig selector: statistical estimation when p is much larger than n, *The Annals of Statistics* **35**: 2313–2351.

Cortes, C. and Vapnik, V. (1995). Support-vector networks, *Machine Learning* **20**: 273–297.

Dantzig, G. (1951). *Linear Programming and Extensions*, Princeton University Press.

Efron, B., Hastie, T., Johnstone, I. and Tibshirani, R. (2004). Least angle regression (with discussion), *The Annals of Statistics* **32**: 407–451.

Fisher, W. D. (1961). A note on curve fitting with minimum deviations by linear programming, *Journal of the American Statistical Association* **56**: 359–362.

Ǵal, T. (1979). *Postoptimal Analyses, Parametric Programming, and Related Topics*, McGraw-Hill International, New York.

Gass, S. and Saaty, T. (1955a). The computational algorithm for the parametric objective function, *Naval Research Logistics Quarterly* **2**: 39–45.

Gass, S. and Saaty, T. (1955b). The parametric objective function (part 2), *Journal of the Operations Research Society of America* **3**: 395–401.

Gill, P. E., Murray, W. and Wright, M. H. (1991). *Numerical Linear Algebra and Optimization*, Addison-Wesley, Reading, MA.

Gunn, S. R. and Kandola, J. S. (2002). Structural modelling with sparse kernels, *Mach. Learning* **48**(1): 137–63.

Hastie, T., Rosset, S., Tibshirani, R. and Zhu, J. (2004). The entire regularization path for the support vector machine, *Journal of Machine Learning Research* **5**: 1391–1415.

Hastie, T., Tibshirani, R. and Friedman, J. (2001). *The Elements of Statistical Learning*, Springer Verlag, New York.

Hoerl, A. and Kennard, R. (1970). Ridge regression: Biased estimation for nonorthogonal problems, *Technometrics* **12**: 55–67.

Karmarkar, N. (1984). A new polynomial-time algorithm for linear programming, *Combinatorica* **4**(4): 373–395.

Kim, S.-J., Koh, K., Lustig, M., Boyd, S. and Gorinevsky, D. (2007). An interior-point method for large-scale l1-regularized least squares, *IEEE Journal on Selected Topics in Signal Processing* **1**(4): 606–617.

Koenker, R. (2005). *Quantile Regression (Econometric Society Monographs)*, Cambridge University Press.

Koenker, R. and Bassett, G. (1978). Regression quantiles, *Econometrica* **1**: 33–50.

Koenker, R. and D'Orey, V. (1987). Algorithm AS 229: computing regression quantiles, *Applied Statistics* **36**: 383–393.

Koenker, R. and D'Orey, V. (1994). Remark on algorithm AS 229: computing dual regression quantiles and regression rank scores, *Applied Statistics* **43**: 410–414.

Koenker, R. and Hallock, K. (2001). Quantile regression, *Journal of Economic Perspectives* **15**: 143–156.

Koh, K., Kim, S.-J. and Boyd, S. (2007). An interior-point method for large-scale l1-regularized logistic regression, *Journal of Machine Learning Research* **8**: 1519–1555.

Lee, Y. and Cui, Z. (2006). Characterizing the solution path of multicategory support vector machine, *Statistica Sinica* **16**: 391–409.

Lee, Y., Kim, Y., Lee, S. and Koo, J.-Y. (2006). Structured Multicategory Support Vector Machine with ANOVA decomposition, *Biometrika* **93**(3): 555–571.

Li, Y. and Zhu, J. (2008). L1-norm quantile regressions, *Journal of Computational and Graphical Statistics* **17**: 163–185.

Lin, Y. and Zhang, H. H. (2006). Component selection and smoothing in multivariate nonparametric regression, *The Annals of Statistics* **34**: 2272–2297.

Mehrotra, S. (1992). On the implementation of a primal-dual interior point method, *SIAM Journal on Optimization* **2**(4): 575–601.

Micchelli, C. and Pontil, M. (2005). Learning the kernel function via regularization, *J. Mach. Learning Res.* **6**: 1099–125.

Murty, K. (1983). *Linear Programming*, Wiley, New York, NY.

Rosset, S. and Zhu, J. (2007). Piecewise linear regularized solution paths, *The Annals of Statistics* **35**(3): 1012–1030.

Saaty, T. and Gass, S. (1954). The parametric objective function (part 1), *Journal of the Operations Research Society of America* **2**: 316–319.

Schölkopf, B. and Smola, A. (2002). *Learning with Kernels - Support Vector Machines, Regularization, Optimization and Beyond*, MIT Press, Cambridge, MA.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso, *Journal of the Royal Statistical Society, Series B* **58**: 267–288.

Vanderbei, R. J. (1997). *Linear Programming: Foundations and Extensions*, Springer.

Vapnik, V. (1998). *Statistical Learning Theory*, Wiley, New York: NY.

Wagner, H. M. (1959). Linear programming techniques for regression analysis, *Journal of the American Statistical Association* **54**: 206–212.

Wahba, G. (1990). *Spline Models for Observational Data*, Vol. 59 of *Applied Mathematics*, SIAM, Philadelphia: PA.

Wang, L. and Shen, X. (2006). Multi-category support vector machines, feature selestion and solution path, *Statistica Sinica* **16**: 617–633.

Wright, M. H. (1992). Interior methods for constrained optimization, *Acta Numerica* **1**: 341–407.

Wright, S. J. (1997). *Primal-Dual Interior-Point Methods*, Society for Industrial Mathematics.

Yao, Y. (2008). *Statistical Applications of Linear Programming for Feature Selection via Regularization Methods*, PhD thesis, The Ohio State University.

Yuan, M. and Lin, Y. (2006). Model selection and estimation in regression with grouped variables, *Journal of the Royal Statistical Society, Series B* **68**: 49–67.

Zhang, H. H. (2006). Variable selection for support vector machines via smoothing spline ANOVA, *Statistica Sinica* **16**(2): 659–674.

Zhang, H., Liu, Y., Wu, Y. and Zhu, J. (2006). *Variable selection for multicategory SVM via sup-norm regularization*, North Carolina State University.

Zhu, J., Rosset, S., Hastie, T. and Tibshirani, R. (2004). 1-norm support vector machines, *in* S. Thrun, L. Saul and B. Schölkopf (eds), *Advances in Neural Information Processing Systems 16*, MIT Press, Cambridge, MA.

Zou, H. and Yuan, M. (2008). The f_∞-norm Support Vector Machine, *Statistica Sinica* **18**: 379–398.