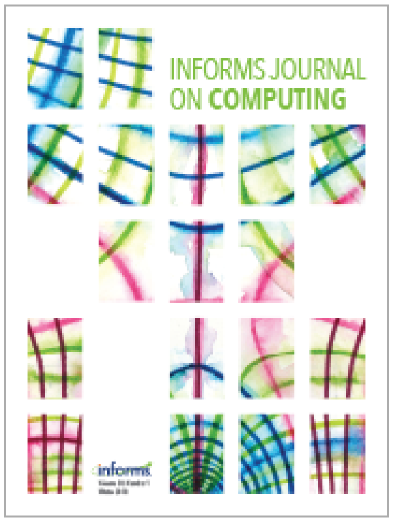
This article was downloaded by: [158.132.13.163] On: 10 April 2025, At: 01:09
Publisher: Institute for Operations Research and the Management Sciences (INFORMS)
INFORMS is located in Maryland, USA



INFORMS Journal on Computing

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

Convex and Nonconvex Risk-Based Linear Regression at Scale

Can Wu, Ying Cui, Donghui Li, Defeng Sun

To cite this article:
Can Wu, Ying Cui, Donghui Li, Defeng Sun (2023) Convex and Nonconvex Risk-Based Linear Regression at Scale. INFORMS Journal on Computing 35(4):797-816. https://doi.org/10.1287/ijoc.2023.1282

Full terms and conditions of use: https://pubsonline.informs.org/Publications/Librarians-Portal/PubsOnLine-Terms-and-Conditions

This article may be used only for the purposes of research, teaching, and/or private study. Commercial use or systematic downloading (by robots or other automatic processes) is prohibited without explicit Publisher approval, unless otherwise noted. For more information, contact permissions@informs.org.

The Publisher does not warrant or guarantee the article's accuracy, completeness, merchantability, fitness for a particular purpose, or non-infringement. Descriptions of, or references to, products or publications, or inclusion of an advertisement in this article, neither constitutes nor implies a guarantee, endorsement, or support of claims made of that product, publication, or service.

Copyright © 2023, INFORMS

Please scroll down for article—it is on subsequent pages

With 12,500 members from nearly 90 countries, INFORMS is the largest international association of operations research (O.R.) and analytics professionals and students. INFORMS provides unique networking and learning opportunities for individual professionals, and organizations of all types and sizes, to better understand and use O.R. and analytics tools and methods to transform strategic visions and achieve better outcomes.
For more information on INFORMS, its publications, membership, or meetings visit http://www.informs.org

https://pubsonline.informs.org/journal/ijoc

INFORMS JOURNAL ON COMPUTING
Vol. 35, No. 4, July–August 2023, pp. 797–816
ISSN 1091-9856 (print), ISSN 1526-5528 (online)

Convex and Nonconvex Risk-Based Linear Regression at Scale

Can Wu,[a,b] **Ying Cui,**[c,*] **Donghui Li,**[a] **Defeng Sun**[b]

[a] School of Mathematical Sciences, South China Normal University, Guangzhou, 510631, China; [b] Department of Applied Mathematics, The Hong Kong Polytechnic University, Hung Hom, Hong Kong; [c] Department of Industrial and Systems Engineering, University of Minnesota, Minneapolis, Minnesota 55455

*Corresponding author

Contact: 2019010105@m.scnu.edu.cn (CW); yingcui@umn.edu, https://orcid.org/0000-0003-4173-5647 (YC); lidonghui@m.scnu.edu.cn (DL); defeng.sun@polyu.edu.hk (DS)

Received: January 9, 2022
Revised: August 31, 2022; December 26, 2022; January 16, 2023; January 19, 2023
Accepted: January 23, 2023
Published Online in Articles in Advance: April 11, 2023

https://doi.org/10.1287/ijoc.2023.1282

Copyright: © 2023 INFORMS

Abstract. The value at risk (VaR) and the conditional value at risk (CVaR) are two popular risk measures to hedge against the uncertainty of data. In this paper, we provide a computational toolbox for solving high-dimensional sparse linear regression problems under either VaR or CVaR measures, the former being nonconvex and the latter convex. Unlike the empirical risk (neutral) minimization models in which the overall losses are decomposable across data, the aforementioned risk-sensitive models have nonseparable objective functions so that typical first order algorithms are not easy to scale. We address this scaling issue by adopting a semismooth Newton-based proximal augmented Lagrangian method of the convex CVaR linear regression problem. The matrix structures of the Newton systems are carefully explored to reduce the computational cost per iteration. The method is further embedded in a majorization–minimization algorithm as a subroutine to tackle the nonconvex VaR-based regression problem. We also discuss an adaptive sieving strategy to iteratively guess and adjust the effective problem dimension, which is particularly useful when a solution path associated with a sequence of tuning parameters is needed. Extensive numerical experiments on both synthetic and real data demonstrate the effectiveness of our proposed methods. In particular, they are about 53 times faster than the commercial package Gurobi for the CVaR-based sparse linear regression with 4,265,669 features and 16,087 observations.

History: Accepted by Antonio Frangioni, Area Editor for Design & Analysis of Algorithms–Continuous.
Funding: This work was supported in part by the NSF, the Division of Computing and Communication Foundations [Grant 2153352], the National Natural Science Foundation of China [Grant 12271187], and the Hong Kong Research Grant Council [Grant 15304019].
Supplemental Material: The software that supports the findings of this study is available within the paper and its Supplemental Information (https://pubsonline.informs.org/doi/suppl/10.1287/ijoc.2023.1282) as well as from the IJOC GitHub software repository (https://github.com/INFORMSJoC/2022.0012) at (http://dx.doi.org/10.5281/zenodo.7483279).

Keywords: risk measures • (conditional) value-at-risk • sparsity • semismooth Newton • augmented Lagrangian • nonconvexity

Downloaded from informs.org by [158.132.13.163] on 10 April 2025, at 01:09 . For personal use only, all rights reserved.

1. Introduction

In statistical learning, the empirical risk minimization that minimizes the average losses over the training data set is a risk-neutral way to learn a statistical model. However, in some situations, decision makers may be concerned more about the losses over the left tail to exclude outliers or the right tail to avoid overfitting. Adopting the terminology in risk management, we call this procedure risk-sensitive statistical learning.

Let A be an $n \times d$ matrix with each row $\boldsymbol{A_i}$ representing the ith observation of d-dimensional input features and $\boldsymbol{b}$ be an n-dimensional vector with each b_i representing the ith output response. The following optimization model can be used to estimate the linear relationship between the input–output pairs:

$$\underset{\boldsymbol{x}\in\mathbb{R}^d}{\text{minimize}}\ \mathcal{R}[(\ell(b_i - \boldsymbol{A_i x}))_{i\in[n]}] + \lambda\, p(\boldsymbol{x}), \tag{1}$$

where $\ell : \mathbb{R} \to \mathbb{R}_+$ is a nonnegative univariate function that measures the individual discrepancy between the true output b_i and the model's predicted output $\boldsymbol{A_i x}$, the function $p : \mathbb{R}^d \to \mathbb{R}$ parameterized by a positive scalar λ is a regularizer that forces a prescribed structure on the solution x such as sparsity, and $\mathcal{R} : \mathbb{R}^n \to \mathbb{R}$ is a risk measure that represents one's risk attitude toward training the model. We use the notation $(z_i)_{i\in[n]}$ to represent the n-dimensional vector $(z_1, \ldots, z_n)^\top$ when each z_i is a scalar. If the decision maker is risk-neutral, the decision maker may simply take the expectation over the empirical probability distribution of the observed data $\{\boldsymbol{A_i}, b_i\}_{i=1,\ldots,n}$ as the

Downloaded from informs.org by [158.132.13.163] on 10 April 2025, at 01:09 . For personal use only, all rights reserved.

function $\mathcal{R}$ such that Problem (1) reduces to the usual regularized empirical risk minimization problem

$$\underset{x\in\mathbb{R}^d}{\text{minimize}}\ \frac{1}{n}\sum_{i=1}^{n}\ell(b_i-\boldsymbol{A_i x})+\lambda\, p(\boldsymbol{x}). \tag{2}$$

There is extensive literature on first and second order methods to solve various combinations of the error function ℓ and the regularizer p. We refer interested readers to the monograph by Hastie et al. (2015) for a comprehensive review of the algorithms and statistical properties for Problem (2).

Different from the risk-neutral approach, the risk-sensitive learning model takes an asymmetric view of the individual losses $\{\ell(b_i-\boldsymbol{A_i x})\}_{i=1,\ldots,n}$ distributed on two tails. In this paper, we are interested in two risk measures: the value at risk (VaR) and the conditional value at risk (CVaR). Suppose that one observes n realizations of a nonnegative random variable Z, denoted as $z_1,\ldots,z_n$, each with probability $1/n$. Assume further that $k:=(1-\alpha)n$ is a positive integer. Then, the VaR (also called the quantile) of Z with respect to its empirical distribution at the confidence level α is defined as

$$\text{VaR}_\alpha(Z):=\underset{c\in\mathbb{R}}{\text{argmin}}\left\{\frac{1}{n}\sum_{i=1}^{n}1_{\{z_i\le c\}}\ge\alpha\right\}=z_{[k+1]}; \tag{3}$$

the CVaR (also called the superquantile) of Z with respect to its empirical distribution at the confidence level α is given by (c.f. Rockafellar and Uryasev 2000)

$$\text{CVaR}_\alpha(Z):=\underset{c\in\mathbb{R}}{\text{minimum}}\left\{c+\frac{1}{(1-\alpha)n}\sum_{i=1}^{n}\max(z_i-c,0)\right\}=\frac{1}{k}\|\boldsymbol{z}\|_{(k)}. \tag{4}$$

In these two formulations, $1_{\{z_i\le c\}}$ is the indicator function of $\{z_i\le c\}$ that takes the value one if $z_i\le c$ and zero otherwise, $z_{[1]}\ge z_{[2]}\ge\cdots\ge z_{[n]}\ge 0$ are the components of the nonnegative vector z arranged in the nonincreasing order, and $\|\boldsymbol{z}\|_{(k)}=\sum_{i=1}^{k}z_{[i]}$ is its Ky–Fan k-norm (Fan 1951, Horn and Johnson 2013). Adopting one of these two risk measures as $\mathcal{R}$ in (1) and taking $p(\boldsymbol{x})=\|\boldsymbol{x}\|_1$, we may obtain the following two optimization problems:

$$\text{VaR-based sparse linear regression}:\quad \underset{x\in\mathbb{R}^d}{\text{minimize}}\quad [(\ell(b_i-\boldsymbol{A_i x}))_{i\in[n]}]_{[k+1]}+\lambda\|\boldsymbol{x}\|_1; \tag{5}$$

$$\text{CVaR-based sparse linear regression}:\quad \underset{x\in\mathbb{R}^d}{\text{minimize}}\quad \frac{1}{k}\|(\ell(b_i-\boldsymbol{A_i x}))_{i\in[n]}\|_{(k)}+\lambda\|\boldsymbol{x}\|_1. \tag{6}$$

The ℓ_1 penalty terms here are used to enforce sparsity of the solutions to identify important features. Different from Problem (2), these two problems concentrate on the right tail of the losses by either minimizing merely the $(k+1)$th largest one or the average over the top k largest ones. In fact, one may consider a more general truncated CVaR-based model

$$\underset{x\in\mathbb{R}^d}{\text{minimize}}\quad \frac{1}{k_1-k_2}\left(\left\|(\ell(b_i-\boldsymbol{A_i x}))_{i\in[n]}\right\|_{(k_1)}-\left\|(\ell(b_i-\boldsymbol{A_i x}))_{i\in[n]}\right\|_{(k_2)}\right)+\lambda\|\boldsymbol{x}\|_1 \tag{7}$$

for some $0\le k_2<k_1\le n$, where the averaged losses from the k_1th largest to the (k_2+1)th largest are taken into account. Usually, the exclusion of the bottom $(n-k_1)$'s losses is because the decision maker is risk-averse so that the decision maker wants to focus only on the bad scenarios with large losses, whereas the top k_2's losses are excluded because they are treated as outliers. Obviously both Problems (5) and (6) are special cases of (7) with proper choices of k_1 and k_2. When $k_1=n$ and $k_2\ge 1$, Problem (7) is risk-seeking because it minimizes the average over the bottom $(n-k_2)$'s smallest losses.

Risk-sensitive learning has attracted attention from various communities. The popular (sparse) least trimmed squares estimation in statistics (Rousseeuw 1984, Wang et al. 2007, Alfons et al. 2013) is a special instance of (7), in which only the k smallest squared residuals are considered to exclude outliers in the data. In operations research, the CVaR-based learning problem (6) is closely connected to the (distributionally) robust optimization. In particular, if ℓ is the absolute value function, then Problem (6) is equivalent to the robust linear regression problem whose ambiguity set is constructed by the k-norm ball (Xu et al. 2010). In machine learning, Problem (7) covers the average top k aggregate loss model (Lyu et al. 2020) and is closely related to the hard example mining approach to train a

Downloaded from informs.org by [158.132.13.163] on 10 April 2025, at 01:09 . For personal use only, all rights reserved.

deep neural network (Bengio et al. 2009, Shrivastava et al. 2016) in which larger weights are assigned to the right tail of the losses. Although risk-sensitive models show promising numerical results in many applications, a highly stable, efficient, and scalable numerical solver is not available to date, and this hinders the popularity of the models to some extent.

Our first goal in this paper is to design an efficient method to solve the convex problem (6) when d is huge (say, millions) and n is large (say, thousands). Compared with the empirical risk minimization problem (2), the difficulty of solving Problem (6) mainly comes from two sources: One is the nonsmoothness of the CVaR function (even if ℓ is smooth); this makes the overall objective function the sum of two nonsmooth parts, which precludes the proximal gradient type methods to solve it. The other is the nonseparability of the CVaR-based loss across data so that one cannot easily obtain an unbiased (sub)gradient using a subset of the data to reduce the computational cost via sampling. The recent paper by Levy et al. (2020) tries to address the latter issue by adopting the biased sampling or the nearly unbiased multilevel Monte Carlo gradient estimator to approximate the subgradient of the CVaR function, in which the loss measurement ℓ is assumed to be continuously differentiable. Their proposed approaches inherit both advantages and disadvantages of the stochastic first order–type methods: that one trades the stability and efficiency for the scalability. When ℓ is the square or the absolute value function, a different way to solve (6) is to reformulate it to a standard quadratic program (QP) or linear program (LP) so that one can directly solve them by calling off-the-shelf QP or LP solvers. In fact, this is one of the approaches adopted by the commercial package Portfolio Safeguard,[1] which relies on the Gurobi to solve the reformulated problems. Although Portfolio Safeguard works well for small-to-medium-sized instances, it cannot handle large instances very efficiently, especially when the ℓ_1-regularizer appears in the objective function.

Keeping both the stability and scalability in mind, we develop a semismooth Newton-based proximal augmented Lagrangian method to solve the dual formulation of Problem (6). The main motivation for us to consider this approach is to take advantage of the potential solution sparsity (which, in particular, holds if $d \gg n$) to reduce the effective dimension of the Newton system. We show that, for each step of the semismooth Newton method, the computational cost to invert the (generalized) Hessian matrix is determined by the cardinality of the support of the previous iterates, which ultimately equals the number of nonzero entries of the solution. The latter value is expected to be much smaller than the true variable dimension d under a reasonable choice of the penalty parameter λ, making our approach scalable in terms of d. We further modify the aforementioned approach so that it can be used as a subroutine to solve the nonconvex Problems (5) and (7), in which the majorization–minimization (MM) algorithm is adopted as the outer loop driver.

Our second contribution of this paper is an iterative screening method to guess and adjust the irrelevant features before the start of the aforementioned optimization process, which is particularly useful when a solution path associated with a sequence of λ's is needed. The existing safe screening methods (El Ghaoui et al. 2012, Wang et al. 2014, Ndiaye et al. 2015, Shibagaki et al. 2016) and heuristic screening methods (Fan and Lv 2008, Tibshirani et al. 2012, Lin et al. 2020) produce (nearly) valid regions to eliminate unnecessary features based on dual feasible solutions. These methods can be overly conservative and retain an unnecessarily large proportion of variables for the problem at the current grid point, especially when the grid is coarse. In contrast, our proposed approach is more aggressive by forcing all entries outside of the support of a given point (usually the solution at the previous grid point) to be zero in the first step. We then adjust the effective space of the model based on the Karush–Kuhn–Tucker (KKT) conditions of the current problem and solve the newly generated restricted problem repeatedly until all KKT conditions are satisfied. Our numerical results show that the average reduced subproblem sizes nearly match the actual number of nonzeros in the optimal solutions.

It is worth mentioning that, although we focus on the linear regression problem in the present paper, the proposed computational framework to deal with the nonseparable risk measures can be easily generalized to other large-scale risk-management problems arising from operations research and financial engineering.

The rest of the paper is organized as follows. In Section 2, we present some background on the proximal mapping and semismooth functions. Section 3 is devoted to the computational framework of the convex CVaR-based linear regression problem (6). We first introduce the proximal augmented Lagrangian method applied to the dual formulation. Subsequently, an efficient and scalable semismooth Newton method is discussed to solve the augmented Lagrangian subproblem, and we show how to take advantage of the potential solution sparsity to efficiently solve the linear Newton equation. In Section 4, we present an adaptive sieving strategy as a preprocess to guess and adjust the support of the primal solution iteratively, which is, in particular, useful to efficiently compute a solution path of the CVaR-based learning problem (6) for a fine grid of λ's. Section 5 is about a majorization–minimization algorithm to solve the nonconvex problem (7), in which the subproblems are solved based on a similar algorithm in Section 3. Extensive numerical experiments on synthetic and real data are presented in Section 6, and they demonstrate the effectiveness of our proposed methods.

Downloaded from informs.org by [158.132.13.163] on 10 April 2025, at 01:09 . For personal use only, all rights reserved.

2. Preliminaries

To proceed, we first list notations used in the later discussion. We write $E_{m\times n}$ as the $m \times n$ matrix of all ones, $\boldsymbol{e_n}(= E_{n\times 1})$ as the n-dimensional vector of all ones, $0_{m\times n}$ as the $m \times n$ matrix of all zeros, and I_n as the $n \times n$ identity matrix. For any given positive integer n, denote $[n] := \{1,2,\ldots,n\}$. For any $\boldsymbol{x} \in \mathbb{R}^m$, we write $\mathrm{Diag}(\boldsymbol{x})$ as the $m \times m$ diagonal matrix whose ith diagonal entry is x_i for $i = 1,2,\ldots,m$. For any given collection of square matrices $\{A_\ell \in \mathbb{R}^{a_\ell \times a_\ell}\}_{\ell=1}^r$, the notation $\mathrm{Diag}(A_1, A_2, \ldots, A_r)$ represents the block diagonal matrix whose ℓth diagonal block is A_ℓ. Generalizing the Ky–Fan k-norm of a nonnegative vector in Section 1, we use the same notation to define the k-norm of any vector $\boldsymbol{z} \in \mathbb{R}^n$ as $\|z\|_{(k)} := \sum_{i=1}^k |z|_{[i]}^{\downarrow}$, where $\boldsymbol{w}^{\downarrow}$ is the vector of entries of $\boldsymbol{w}$ that are arranged in the nonincreasing order $w_{[1]}^{\downarrow} \geq \cdots \geq w_{[n]}^{\downarrow}$ for any $\boldsymbol{w} \in \mathbb{R}^n$.

Let $f : \mathbb{R}^n \to (-\infty, +\infty]$ be a given closed proper convex function. The Moreau envelope of f at any $\boldsymbol{x} \in \mathbb{R}^n$ is defined as the value function $e_f(\boldsymbol{x}) := \mathrm{minimum}_{\boldsymbol{y}\in\mathbb{R}^n}\{f(\boldsymbol{y}) + 1/2\|\boldsymbol{y} - \boldsymbol{x}\|^2\}$. The single-valued optimal solution mapping of this problem, which is usually called the proximal point mapping of f, is denoted as Prox_f. It is known that the Moreau envelop of any proper closed convex function is continuously differentiable with the gradient

$$\nabla e_f(\boldsymbol{x}) = \boldsymbol{x} - \mathrm{Prox}_f(\boldsymbol{x}), \quad \boldsymbol{x} \in \mathbb{R}^n. \tag{8}$$

Detailed properties of the Moreau envelope and the proximal mapping can be found in Rockafellar and Wets (1998, chapter 1.G). In particular, the Moreau identity $\boldsymbol{x} = \mathrm{Prox}_f(\boldsymbol{x}) + \mathrm{Prox}_{f^*}(\boldsymbol{x})$ plays an important role in our algorithmic development, where $f^*(\boldsymbol{y}) := \sup_{\boldsymbol{x}\in\mathbb{R}^n}\{\langle \boldsymbol{y}, \boldsymbol{x}\rangle - f(\boldsymbol{x})\}$ denotes the conjugate function of f. It is known that (Bhatia 1997, exercises IV.1.18) the dual norm of the vector k-norm is given by

$$\|\boldsymbol{x}\|_{(k)^*} = \max\left\{\|\boldsymbol{x}\|_\infty, \frac{1}{k}\|\boldsymbol{x}\|_1\right\}, \quad \boldsymbol{x} \in \mathbb{R}^n.$$

Hence, for any $r > 0$,

$$\mathrm{Prox}_{r\|\cdot\|_{(k)}}(\boldsymbol{x}) = \boldsymbol{x} - \mathrm{Prox}_{\delta_{\mathcal{B}^r_{(k)^*}}}(\boldsymbol{x}) = \boldsymbol{x} - \Pi_{\mathcal{B}^r_{(k)^*}}(\boldsymbol{x}),$$

where $\mathcal{B}^r_{(k)^*}$ is the $(k)^*$-norm ball with the radius r given by $\mathcal{B}^r_{(k)^*} := \{\boldsymbol{x} \in \mathbb{R}^n \,|\, \|\boldsymbol{x}\|_\infty \leq r, \|\boldsymbol{x}\|_1 \leq kr\}$. In fact, it is known from Helgason et al. (1980) that the projection onto this ball has an explicit expression $\Pi_{\mathcal{B}^r_{(k)^*}}(\boldsymbol{x}) = \mathrm{sign}(\boldsymbol{x}) \circ \boldsymbol{y}$, where the nonnegative vector y is defined as

$$\boldsymbol{y} = \begin{cases} \boldsymbol{y}(\lambda^*) & \text{if } \sum_{i=1}^n y_i(\lambda^*) = kr \\ \min\{r\boldsymbol{e_n}, |\boldsymbol{x}|\} & \text{otherwise} \end{cases}$$

with the vector $\boldsymbol{y}(\lambda^*)$ given by $y_i(\lambda^*) := \max\{\min(|x_i| - \lambda^*, r), 0\}$ for $i \in [n]$, and $\lambda^* \in \mathbb{R}$ is a solution of the equation $\sum_{i=1}^n y_i(\lambda^*) = kr$. The value of λ^* can be computed by the breakpoint searching algorithm with $\mathcal{O}(n\log n)$ complexity (Held et al. 1974, Helgason et al. 1980).[2] One can also make use of median of breakpoint subsets to further reduce the complexity to $\mathcal{O}(n)$ (Brucker 1984, Pardalos and Kovoor 1990).

Next, we introduce the concepts of semismoothness and generalized Jacobian. One may find these definitions in Facchinei and Pang (2007, chapter 7). Suppose that $F : \mathcal{O} \subset \mathbb{R}^n \to \mathbb{R}^m$ is a locally Lipschitz continuous function on an open set $\mathcal{O}$. Based on Rademacher's theorem, F is then (Fréchet) differentiable almost everywhere in $\mathcal{O}$. Let $D_F := \{\boldsymbol{x} \in \mathbb{R}^n \,|\, F \text{ is differentiable at } \boldsymbol{x}\}$ and denote $F'(\boldsymbol{x})$ as the derivative of F at $\boldsymbol{x} \in D_F$. Then, the Bouligand subdifferential of F at $\boldsymbol{x} \in \mathcal{O}$ is defined as

$$\partial_B F(\boldsymbol{x}) := \left\{\lim_{i\to\infty} F'(\boldsymbol{x}^i) \,\middle|\, \boldsymbol{x}^i \in D_F, \boldsymbol{x}^i \to \boldsymbol{x}\right\}.$$

The Clarke generalized Jacobian of F at $\boldsymbol{x} \in \mathcal{O}$ is the convex hull of $\partial_B F(\boldsymbol{x})$, written as $\partial F(\boldsymbol{x}) := \mathrm{conv}\{\partial_B F(\boldsymbol{x})\}$. The function F is called semismooth at a point $\boldsymbol{x} \in \mathcal{O}$ if F is directionally differentiable at $\boldsymbol{x}$ and for any $\boldsymbol{h} \to 0$ and $V \in \partial F(\boldsymbol{x} + \boldsymbol{h})$,

$$F(\boldsymbol{x} + \boldsymbol{h}) - F(\boldsymbol{x}) - V\boldsymbol{h} = o(\|\boldsymbol{h}\|).$$

If $o(\|\boldsymbol{h}\|)$ is replaced by $O(\|\boldsymbol{h}\|^2)$ in the preceding equation, then F is called strongly semismooth at $\boldsymbol{x} \in \mathcal{O}$.

Downloaded from informs.org by [158.132.13.163] on 10 April 2025, at 01:09 . For personal use only, all rights reserved.

3. An Algorithm for the CVaR-Based Learning

In this section, we propose a semismooth Newton-based augmented Lagrangian method to solve the convex CVaR-based learning problem (6) and discuss how to leverage the solution sparsity to reduce the computation cost so that the proposed algorithm is scalable in terms of both the number of samples n and the dimension of the features d.

For notational simplicity, we denote $f(z) := \|(\ell(z_i))_{i\in[n]}\|_{(k)}$ and $h(x) := \|x\|_1$ for any $z \in \mathbb{R}^n$ and $x \in \mathbb{R}^d$. Then, Problem (6) can be equivalently written as

$$\underset{x\in\mathbb{R}^d,\, z\in\mathbb{R}^n}{\text{minimize}}\{f(z) + \lambda h(x) \,|\, Ax - z = b\}, \tag{9}$$

whose Lagrangian dual problem is given by

$$\underset{u\in\mathbb{R}^n}{\text{maximize}} - \{g(u) := (\lambda h)^*(-A^\top u) + f^*(u) + \langle u, b\rangle\}. \tag{10}$$

Notice that the objective function of Problem (6) is real-valued convex and lower level bounded. We then know from Rockafellar and Wets (1998, theorem 1.9) that its optimal value is finite and solution set is nonempty and so is that of (9). Because there are only linear constraints in (9), it further follows from Bertsekas et al. (2003, proposition 6.4.2) that the optimal values of the primal problem (9) and dual problem (10) are the same.

3.1. A Proximal Augmented Lagrangian Method for the Dual Problem

We first derive the proximal augmented Lagrangian function for the (unconstrained) dual problem (10) by the general theory in Rockafellar and Wets (1998, definition 11.45 and example 11.57). Define a proper closed convex function $\tilde{g} : \mathbb{R}^n \times \mathbb{R}^d \times \mathbb{R}^n \to (-\infty, +\infty]$ as

$$\tilde{g}(u, \xi_1, \xi_2) := (\lambda h)^*(-A^\top u + \xi_1) + f^*(u + \xi_2) + \langle u, b\rangle.$$

Obviously, $\tilde{g}(u, 0, 0) = g(u)$ for any $u \in \mathbb{R}^n$, where g is the negative dual objective function in (10). The Lagrangian function $L : \mathbb{R}^n \times \mathbb{R}^d \times \mathbb{R}^n \to (-\infty, +\infty]$ of (10) is given by

$$\begin{aligned} L(u; x, z) &:= \inf_{(\xi_1,\xi_2)\in\mathbb{R}^d\times\mathbb{R}^n} \{\tilde{g}(u, \xi_1, \xi_2) - \langle x, \xi_1\rangle - \langle z, \xi_2\rangle\} \\ &= -\lambda h(x) - f(z) - \langle x, A^\top u\rangle + \langle z, u\rangle + \langle u, b\rangle. \end{aligned}$$

Given a positive scalar σ, the augmented Lagrangian function is then given by

$$\begin{aligned} \overline{L}_\sigma(u; x, z) &:= \sup_{(\xi_1,\xi_2)\in\mathbb{R}^d\times\mathbb{R}^n} \left\{ L(u; \xi_1, \xi_2) - \frac{1}{2\sigma}\|\xi_1 - x\|^2 - \frac{1}{2\sigma}\|\xi_2 - z\|^2 \right\} \\ &= \frac{1}{\sigma}\left[\frac{1}{2}\|x - \sigma A^\top u\|^2 - e_{\sigma\lambda h}(x - \sigma A^\top u)\right] - \frac{1}{2\sigma}\|x\|^2 \\ &\quad + \frac{1}{\sigma}\left[\frac{1}{2}\|z + \sigma u\|^2 - e_{\sigma f}(z + \sigma u)\right] - \frac{1}{2\sigma}\|z\|^2 + \langle u, b\rangle. \end{aligned}$$

Based on the augmented Lagrangian function, we present the proximal augmented Lagrangian method to solve the dual problem (10) in Algorithm 1.

Algorithm 1 (A Proximal Augmented Lagrangian Method for the Dual Problem (10))

Initialization: Given are a positive nondecreasing sequence $\{\sigma_t\}_{t\ge 0}$ and a positive nonincreasing sequence $\{\tau_t\}_{t\ge 0}$. Choose $(u^0, x^0, z^0) \in \mathbb{R}^n \times \mathbb{R}^d \times \mathbb{R}^n$. Set $t = 0$. Repeat the following steps until a proper stopping criterion is satisfied by the sequence $\{(u^t, x^t, z^t)\}$.

1: Step 1. Compute an approximate solution

$$u^{t+1} \approx \underset{u\in\mathbb{R}^n}{\arg\min}\left\{\varphi_t(u) := \overline{L}_{\sigma_t}(u; x^t, z^t) + \frac{\tau_t}{2\sigma_t}\|u - u^t\|^2\right\}. \tag{11}$$

2: Step 2. Update the multipliers

$$(x^{t+1}, z^{t+1}) = \left(\text{Prox}_{\sigma_t\lambda h}(x^t - \sigma_t A^\top u^{t+1}),\ \text{Prox}_{\sigma_t f}(z^t + \sigma_t u^{t+1})\right).$$

Downloaded from informs.org by [158.132.13.163] on 10 April 2025, at 01:09 . For personal use only, all rights reserved.

The global convergence and the local superlinear convergence rate of the proximal augmented Lagrangian method are well-studied in the existing literature, see, for example, the seminal work of Rockafellar (1976). For completeness, we briefly discuss proper stopping criteria of the subproblem computation in (11) such that the overall algorithm (a) converges and (b) in an asymptotically superlinear rate.

Following the general framework in Rockafellar (1976), we consider the maximal monotone operator $\mathcal{T}$ induced by the subgradient of the convex–concave Lagrangian function

$$\begin{aligned}\mathcal{T}(\boldsymbol{u},\boldsymbol{x},\boldsymbol{z}) &:= \{(\boldsymbol{u}',\boldsymbol{x}',\boldsymbol{z}')\,|\,(\boldsymbol{u}',-\boldsymbol{x}',-\boldsymbol{z}')\in\partial L(\boldsymbol{u};\boldsymbol{x},\boldsymbol{z})\}\\ &= \{(\boldsymbol{u}',\boldsymbol{x}',\boldsymbol{z}')\,|\,\boldsymbol{u}'=\boldsymbol{z}-A\boldsymbol{x}+\boldsymbol{b},\boldsymbol{x}'\in A^\top\boldsymbol{u}+\partial(\lambda h)(\boldsymbol{x}),\boldsymbol{z}'\in-\boldsymbol{u}+\partial f(\boldsymbol{z})\}.\end{aligned}$$

Denote a block diagonal matrix

$$\mathcal{M}_t := \mathrm{Diag}(\tau_t\, I_n, I_d, I_n) \succ 0. \tag{12}$$

We consider the following stopping criteria adopted in Li et al. (2020) that are implementable conditions based on the general principles in Rockafellar (1976):

$$\|\nabla\varphi_t(\boldsymbol{u}^{t+1})\| \le \frac{\min(\sqrt{\tau_t},1)}{\sigma_t}\varepsilon_t, \tag{A}$$

$$\|\nabla\varphi_t(\boldsymbol{u}^{t+1})\| \le \frac{\delta_t\min(\sqrt{\tau_t},1)}{\sigma_t}\|(\boldsymbol{u}^{t+1},\boldsymbol{x}^{t+1},\boldsymbol{z}^{t+1})-(\boldsymbol{u}^t,\boldsymbol{x}^t,\boldsymbol{z}^t)\|_{\mathcal{M}_t}, \tag{B}$$

where $\{\varepsilon_t\}_{t\ge0}$ and $\{\delta_t\}_{t\ge0}$ are positive summable sequences with each $\delta_t<1$. In the original work of Rockafellar (1976), the local convergence rate is established under the Lipschitz continuity of $\mathcal{T}$. This condition is not easily satisfied by (9) because it, in particular, requires the uniqueness of the primal–dual solution pair. There are extensive works to relax this Lipschitz continuity assumption in the convergence rate analysis of the augmented Lagrangian method. We refer readers to the review paper by Cui et al. (2021) and references therein for recent development on this topic. We state the global convergence and the local asymptotically superlinear convergence rate of Algorithm 1 based on the results in Li et al. (2020). To proceed, we denote $\mathrm{S}_{\mathrm{KKT}}$ as the set of all KKT pairs $(\overline{\boldsymbol{u}},\overline{\boldsymbol{x}},\overline{\boldsymbol{z}})$ satisfying $0\in\mathcal{T}(\overline{\boldsymbol{u}},\overline{\boldsymbol{x}},\overline{\boldsymbol{z}})$ and write $\mathrm{dist}_M(\boldsymbol{x},S)$ as the distance of a point x to a closed convex set S in the M-norm for some positive definite matrix M, that is, $\mathrm{dist}_M(\boldsymbol{x},S):=\inf_{\boldsymbol{y}\in S}\sqrt{(\boldsymbol{x}-\boldsymbol{y})^\top M(\boldsymbol{x}-\boldsymbol{y})}$. The subscript M is omitted if it is the identity matrix.

Theorem 1. *Suppose that $\tau_t\downarrow\tau_\infty>0$ and $\sigma_t\uparrow\sigma_\infty\le\infty$. (a) Let $\{(\boldsymbol{u}^t,\boldsymbol{x}^t,\boldsymbol{z}^t)\}$ be the sequence generated by Algorithm 1 under the stopping criterion (A). Then, $\{(\boldsymbol{u}^t,\boldsymbol{x}^t,\boldsymbol{z}^t)\}$ is bounded, and $\{\boldsymbol{u}^t\}$ and $\{\boldsymbol{x}^t\}$ converge to optimal solutions of the dual problem* (10) *and the primal problem* (9), *respectively. (b) Let $\{(\boldsymbol{u}^t,\boldsymbol{x}^t,\boldsymbol{z}^t)\}$ be the sequence generated by Algorithm 1 under both stopping criteria (A) and (B). Assume in addition that, for any $r>0$, there exists $\kappa>0$ such that*

$$\mathrm{dist}\,((\boldsymbol{u},\boldsymbol{x},\boldsymbol{z}),S_{KKT})\le\kappa\,\mathrm{dist}\,(0,\mathcal{T}(\boldsymbol{u},\boldsymbol{x},\boldsymbol{z}))$$

for all $(\boldsymbol{u},\boldsymbol{x},\boldsymbol{z})$ satisfying $\mathrm{dist}\,((\boldsymbol{u},\boldsymbol{x},\boldsymbol{z}),S_{KKT}))\le r$. *Then, for any $t\ge0$,*

$$\mathrm{dist}_{\mathcal{M}_{t+1}}((\boldsymbol{u}^{t+1},\boldsymbol{x}^{t+1},\boldsymbol{z}^{t+1}),S_{KKT})\le\mu_t\,\mathrm{dist}_{\mathcal{M}_t}((\boldsymbol{u}^t,\boldsymbol{x}^t,\boldsymbol{z}^t),S_{KKT}),$$

where $\mathcal{M}_t$ is given in (12) *and*

$$\mu_t:=\frac{1}{1-\delta_t}\left(\delta_t+\frac{(1+\delta_t)\kappa\gamma_t}{\sqrt{\sigma_t^2+\kappa^2\gamma_t^2}}\right)\to\mu_\infty:=\frac{\kappa\gamma_\infty}{\sqrt{\sigma_\infty^2+\kappa^2\gamma_\infty^2}}<1\quad \text{as } t\to\infty$$

with $\gamma_t:=\max(\tau_t,1)$ and $\gamma_\infty=\max(\tau_\infty,1)$.

The rate is said to be asymptotically superlinear because $\mu_\infty=0$ if $\sigma_\infty=\infty$. In practice, one may take a moderate value of μ to get a decent linear convergence rate. The additional assumption in part (b) is a local error bound condition of the KKT solution mapping that is weaker than the so-called Lipschitz continuity of the set valued mapping $\mathcal{T}^{-1}$ employed in the original work of Rockafellar (1976). Notice that the k-norm function (including the ℓ_1 norm as a special case) is piecewise linear. From Sun (1986, proposition 2.24) and the fact that the sum of two polyhedral multifunctions is also polyhedral, we know that $\mathcal{T}$ is a polyhedral multifunction if the loss function ℓ is convex piecewise linear quadratic. Therefore, for both the quadratic loss or the absolute value loss in Problem (9), the associated operator $\mathcal{T}$ satisfies the assumed condition in part (b); see Li et al. (2020, lemma 2.4 and remark 1) for a formal proof.

Downloaded from informs.org by [158.132.13.163] on 10 April 2025, at 01:09 . For personal use only, all rights reserved.

3.2. A Semismooth Newton Algorithm for Subproblem (11)

Although the proximal augmented Lagrangian method enjoys the superlinear convergence rate for the ℓ_1-regularized CVaR-based minimization problem, one has to solve a nontrivial convex subproblem (11) for each iteration. In this section, we discuss how to design a semismooth Newton method for solving this subproblem efficiently.

Notice that $\varphi_t(\cdot)$ defined in (11) is a strongly convex and smooth function. It follows from (8) that the unique optimal solution of $\varphi_t(\cdot)$ can be computed via the following nonlinear equation:

$$\nabla\varphi_t(\boldsymbol{u}) = -A\mathrm{Prox}_{\sigma_t\lambda h}(\boldsymbol{x}^t - \sigma_t A^\top \boldsymbol{u}) + \mathrm{Prox}_{\sigma_t f}(\boldsymbol{z}^t + \sigma_t\boldsymbol{u}) + \boldsymbol{b} + \frac{\tau_t}{\sigma_t}(\boldsymbol{u} - \boldsymbol{u}^t) = 0. \tag{13}$$

Although the function $\nabla\varphi_t(\cdot)$ is not differentiable because of the nonsmoothness of the proximal mappings $\mathrm{Prox}_{\sigma_t\lambda h}$ and $\mathrm{Prox}_{\sigma_t f}$, it is actually piecewise affine if f is piecewise linear quadratic (Rockafellar and Wets 1998, proposition 12.30). Because every piecewise affine mapping is strongly semismooth (Facchinei and Pang 2007, proposition 7.4.4), we know that the gradient mapping $\nabla\varphi_t(\cdot)$ is strongly semismooth, and the semismooth Newton method is, thus, applicable here.

Consider the following set-valued function $\hat{\partial}^2\varphi_t : \mathbb{R}^n \rightrightarrows \mathbb{R}^{n\times n}$:

$$\hat{\partial}^2\varphi_t(\boldsymbol{u}) := \sigma_t A\partial\mathrm{Prox}_{\sigma_t\lambda h}(\boldsymbol{x}^t - \sigma_t A^\top\boldsymbol{u})A^\top + \sigma_t\partial\mathrm{Prox}_{\sigma_t f}(\boldsymbol{z}^t + \sigma_t\boldsymbol{u}) + \frac{\tau_t}{\sigma_t}I_n, \quad u \in \mathbb{R}^n,$$

where $\partial\mathrm{Prox}_{\sigma_t\lambda h}(\cdot)$ and $\partial\mathrm{Prox}_{\sigma_t f}(\cdot)$ are the Clarke generalized Jacobian of $\mathrm{Prox}_{\sigma_t\lambda h}(\cdot)$ and $\mathrm{Prox}_{\sigma_t f}(\cdot)$, respectively. It follows from Hiriart-Urruty et al. (1984, theorem 2.2) that

$$\hat{\partial}^2\varphi_t(\boldsymbol{u})\boldsymbol{\xi} = \partial^2\varphi_t(\boldsymbol{u})\boldsymbol{\xi}, \quad \forall\boldsymbol{\xi}\in\mathbb{R}^n,$$

where $\partial^2\varphi_t(\boldsymbol{u})$ denotes the generalized Hessian of φ_t at $\boldsymbol{u}$, that is, the Clarke generalized Jacobian of $\nabla\varphi_t$ at $\boldsymbol{u}$. The framework of the semismooth Newton method for solving (11) is demonstrated in Algorithm 2.

Algorithm 2 (A Semismooth Newton Method for Subproblem (11))

```
Initialization: Choose positive scalars $\mu \in (0,1/2)$, $\overline{\tau} \in (0,1]$, $\overline{\gamma}, \delta \in (0,1)$ and an initial point $\boldsymbol{u}^{t,0} \in \mathbb{R}^n$. Set $i = 0$. Execute the following steps until the stopping criteria (A) and/or (B) in the augmented Lagrangian method at $\boldsymbol{u}^{t,i+1}$ are satisfied.
1: Step 1 (finding the semismooth Newton direction). Choose $V^i \in \partial\mathrm{Prox}_{\sigma_t\lambda h}(\boldsymbol{x}^t - \sigma_t A^\top\boldsymbol{u}^{t,i})$ and $W^i \in \partial\mathrm{Prox}_{\sigma_t f}(\boldsymbol{z}^t + \sigma_t\boldsymbol{u}^{t,i})$. Let $U^i := \sigma_t AV^iA^\top + \sigma_t W^i + \tau_t\sigma_t^{-1}I_n$. Apply the direct method or the conjugate gradient method to find an approximate solution $\mathbf{dir}^i \in \mathbb{R}^n$ of the following linear equation
       $U^i\mathbf{dir}^i = -\nabla\varphi_t(\boldsymbol{u}^{t,i})$   (14)
   such that $\|U^i\,\mathbf{dir}^i + \nabla\varphi_t(\boldsymbol{u}^{t,i})\| \le \min(\overline{\gamma}, \|\nabla\varphi_t(\boldsymbol{u}^{t,i})\|^{1+\overline{\tau}})$.
2: Step 2 (line search). Set $\alpha_i = \delta^{m_i}$, where $m_i$ is the first nonnegative integer $m$ for which
       $\varphi_t(\boldsymbol{u}^{t,i} + \delta^m\mathbf{dir}^i) \le \varphi_t(\boldsymbol{u}^{t,i}) + \mu\delta^m\langle\nabla\varphi_t(\boldsymbol{u}^{t,i}), \mathbf{dir}^i\rangle$.
3: Step 3. Set $\boldsymbol{u}^{t,i+1} = \boldsymbol{u}^{t,i} + \alpha_i\mathbf{dir}^i$ and $i \leftarrow i+1$.
```

The global convergence and the convergence rate of the semismooth Newton method are extensively studied in the existing literature; see, for example, Zhao et al. (2010, proposition 3.3 and theorem 3.4) and Li et al. (2018, theorem 3). For completeness, we repeat the results as follows.

Theorem 2. *Let $\{\boldsymbol{u}^{t,i}\}_{i\ge 0}$ be the sequence generated by Algorithm 2. Then, $\{\boldsymbol{u}^{t,i}\}$ converges to the unique optimal solution $\overline{\boldsymbol{u}}^{t+1}$ of Subproblem (11). In addition,*

$$\|\boldsymbol{u}^{t,i+1} - \overline{\boldsymbol{u}}^{t+1}\| = O(\|\boldsymbol{u}^{t,i} - \overline{\boldsymbol{u}}^{t+1}\|^{1+\overline{\tau}}),$$

for all i sufficiently large, where $\overline{\tau} \in (0,1]$ is the constant used in Algorithm 2.

3.3. Solving the Linear Equation (14)

The main computational and storage burdens of the semismooth Newton method discussed in the preceding section are to solve the $n \times n$ dimensional linear Equation (14). Consider the ith iterate of the semismooth Newton method. For brevity, we write $\dot{\boldsymbol{u}} := \boldsymbol{u}^{t,i} \in \mathbb{R}^n$, $\omega_1(\dot{\boldsymbol{u}}) := \boldsymbol{x}^t - \sigma_t A^\top\dot{\boldsymbol{u}} \in \mathbb{R}^d$, $\omega_2(\dot{\boldsymbol{u}}) := \boldsymbol{z}^t + \sigma_t\dot{\boldsymbol{u}} \in \mathbb{R}^n$ and parameters $\sigma := \sigma_t$, $\tau := \tau_t$.

Downloaded from informs.org by [158.132.13.163] on 10 April 2025, at 01:09 . For personal use only, all rights reserved.

Then, the linear system (14) takes the following form:

$$\left(\sigma AVA^\top + \sigma W + \frac{\tau}{\sigma} I_n\right)\mathbf{dir} = \mathbf{rhs}, \tag{15}$$

where $V \in \partial \text{Prox}_{\sigma\lambda\|\cdot\|_1}(\omega_1(\dot{\boldsymbol{u}}))$, $W \in \partial\text{Prox}_{\sigma f}(\omega_2(\dot{\boldsymbol{u}}))$ and rhs $:= -\nabla\varphi_t(\dot{\boldsymbol{u}})$.

In this part, we take $\ell(t) = |t|$ as the loss function in (9) as an example to show how to explore the structures of matrices V and W so that the linear Equation (15) can be solved efficiently. Notice that, for this case, the function $f : \mathbb{R}^n \to \mathbb{R}$ reduces to the Ky–Fan k-norm for a vector in $\mathbb{R}^n$ (not necessarily nonnegative).

3.3.1. The Term *AVA*$^\top$.

We take the following special element V from the generalized Jacobian set $\partial\text{Prox}_{\sigma\lambda\|\cdot\|_1}(\omega_1(\dot{\boldsymbol{u}}))$:

$$V = \text{Diag}(v) \quad \text{with} \quad v_j = \begin{cases} 0 & \text{if} \quad |(\omega_1(\dot{\boldsymbol{u}}))_j| \le \sigma\lambda \\ 1 & \text{otherwise,} \end{cases} \qquad j = 1, 2, \ldots, d.$$

Denote the active index set $\mathcal{J} := \{j \in [d] \,|\, v_j = 1\}$ and its cardinality $s = |\mathcal{J}|$. We have

$$AVA^\top = (AV)(AV)^\top = A_{\mathcal{J}}A_{\mathcal{J}}^\top \in \mathbb{R}^{n\times n},$$

where $A_{\mathcal{J}} \in \mathbb{R}^{n\times s}$ represents the submatrix of A obtained by removing all the columns of A not in $\mathcal{J}$. Therefore, the cost of computing $AVA^\top$ is only $O(n^2 s)$, which is much smaller than the naive matrix multiplication $O(n^2 d)$ when $s \ll d$.

3.3.2. The Matrix *W*.

Let $P(\omega_2(\dot{\boldsymbol{u}})) = |\omega_2(\dot{\boldsymbol{u}})|^{\downarrow}$ with a proper signed permutation matrix $P \in \mathbb{R}^{n\times n}$ and $M \in \partial\Pi_{\mathcal{B}^{\sigma}_{(k)^*}}(P(\omega_2(\dot{\boldsymbol{u}})))$. It follows from the general formula for the generalized Jacobian of $\partial\Pi_{\mathcal{B}^{\sigma}_{(k)^*}}$ stated in Online Appendix A and the Moreau identity that

$$W = I_n - P^\top MP = P^\top(I_n - M)P \in \partial\text{Prox}_{\sigma\|\cdot\|_{(k)}}(\omega_2(\dot{\boldsymbol{u}})).$$

3.3.3. Solving Equation (15).

With $D := \sigma W + \tau\sigma^{-1} I_n$, we can rewrite Equation (15) in the following form

$$(\sigma A_{\mathcal{J}}A_{\mathcal{J}}^\top + D)\mathbf{dir} = \mathbf{rhs}. \tag{16}$$

In our implementation, this equation is solved via different methods based on the values of n and s: (a) If n and s are both very large, we solve the equation by the conjugate gradient method.[3] (b) If $s \ge \theta_1 n$ for some $\theta_1 \in (0, 1]$, one may solve the equation via the Cholesky factorization. (c) If n is large but s is very small (which could especially happen when the iterate is close to a true sparse solution), one may solve (16) efficiently in the following way. Based on the Sherman–Morrison–Woodbury formula, the solution dir of (16) can be computed as

$$\begin{aligned}\mathbf{dir} &= D^{-1}\mathbf{rhs} - D^{-1}A_{\mathcal{J}}(\sigma^{-1}I_s + A_{\mathcal{J}}^\top D^{-1}A_{\mathcal{J}})^{-1}A_{\mathcal{J}}^\top(D^{-1}\mathbf{rhs}) \\ &= D^{-1}\mathbf{rhs} - D^{-1}A_{\mathcal{J}}T^{-1}A_{\mathcal{J}}^\top(D^{-1}\mathbf{rhs}),\end{aligned} \tag{17}$$

where $T := \sigma^{-1}I_s + A_{\mathcal{J}}^\top D^{-1}A_{\mathcal{J}} \in \mathbb{R}^{s\times s}$. Notice that the main computational cost of Equation (17) is to calculate D^{-1} and T^{-1}. In the following, we show in detail that utilizing the special structure of W and the Sherman–Morrison–Woodbury formula, we can compute the inverse of D at a low computational cost. For simplicity, we denote $\boldsymbol{u} := P(\omega_2(\dot{\boldsymbol{u}}))$, $\overline{\boldsymbol{u}} = \Pi_{\mathcal{B}^{\sigma}_{(k)^*}}(\boldsymbol{u})$, $\delta := \sigma + \tau\sigma^{-1}$, $\varpi := (\tau + \sigma^2)^{-1}\sigma^2$ and $r := \sigma$. In addition, we write

$$\alpha := \{i \in [n] \,|\, \overline{u}_i = r\}, \quad \beta := \{i \in [n] \,|\, 0 < \overline{u}_i < r\}, \quad \gamma := [n] \setminus (\alpha \cup \beta).$$

Case 1: $\boldsymbol{u} = \overline{\boldsymbol{u}}$. In this case, $M = I_n$ so that $W = P^\top(I_n - M)P = 0_{n\times n}$ and $D = \tau\sigma^{-1}I_n$, which yields that $D^{-1} = \sigma\tau^{-1}I_n$. Thus, the computational cost of $A_{\mathcal{J}}^\top D^{-1}A_{\mathcal{J}}$ is $O(s^2 n)$.

Case 2: $\boldsymbol{u} \ne \overline{\boldsymbol{u}}, \|\overline{\boldsymbol{u}}\|_\infty = r, \|\overline{\boldsymbol{u}}\|_1 < kr$. We have $W = P^\top(I_n - M)P = P_\alpha^\top P_\alpha$, where $P_\alpha \in \mathbb{R}^{|\alpha|\times n}$ represents the submatrix of P obtained by removing all the rows of P not in α. Thus, $D = \sigma P_\alpha^\top P_\alpha + \tau\sigma^{-1}I_n$ and

$$D^{-1} = \begin{cases} \dfrac{1}{\delta}\left(I_n + \dfrac{\sigma^2}{\tau}P_{\beta\cup\gamma}^\top P_{\beta\cup\gamma}\right) & \text{if } |\alpha| \ge \theta_2|\beta\cup\gamma| \\ \dfrac{\sigma}{\tau}(I_n - \varpi P_\alpha^\top P_\alpha) & \text{if } |\alpha| < \theta_2|\beta\cup\gamma|, \end{cases}$$

which implies that the cost of $A_{\mathcal{J}}^\top D^{-1}A_{\mathcal{J}}$ is $O(s^2(n + |\beta\cup\gamma|))$ when $|\alpha| \ge \theta_2|\beta\cup\gamma|$ and $O(s^2(n + |\alpha|))$ when $|\alpha| < \theta_2|\beta\cup\gamma|$ with a constant $\theta_2 \in (0, 1]$, respectively.

Downloaded from informs.org by [158.132.13.163] on 10 April 2025, at 01:09 . For personal use only, all rights reserved.

Case 3: $\boldsymbol{u} \neq \overline{\boldsymbol{u}}, \|\overline{\boldsymbol{u}}\|_\infty < r, \|\overline{\boldsymbol{u}}\|_1 = kr$. Then,

$$W = P^\top \mathrm{Diag}\left(\frac{1}{|\alpha\cup\beta|}E_{|\alpha\cup\beta|\times|\alpha\cup\beta|}, I_\gamma\right)P = \frac{1}{|\alpha\cup\beta|}\left(\boldsymbol{e}^\top_{|\alpha\cup\beta|}P_{\alpha\cup\beta}\right)^\top\left(\boldsymbol{e}^\top_{|\alpha\cup\beta|}P_{\alpha\cup\beta}\right) + P_\gamma^\top P_\gamma.$$

Hence,

$$D = \sigma\left[\frac{1}{|\alpha\cup\beta|}\left(\boldsymbol{e}^\top_{|\alpha\cup\beta|}P_{\alpha\cup\beta}\right)^\top\left(\boldsymbol{e}^\top_{|\alpha\cup\beta|}P_{\alpha\cup\beta}\right) + P_\gamma^\top P_\gamma\right] + \tau\sigma^{-1}I_n.$$

Subcase 3.1: $|\alpha\cup\beta| \le \theta_2|\gamma|$. One may obtain that

$$D^{-1} = \delta^{-1}[I_n - V_3^\top(\delta\sigma^{-1}\Sigma_3^{-1} + V_3V_3^\top)^{-1}V_3],$$

where $V_3^\top = ((\boldsymbol{e}^\top_{|\alpha\cup\beta|}P_{\alpha\cup\beta})^\top, P^\top_{\alpha\cup\beta}) \in \mathbb{R}^{n\times(1+|\alpha\cup\beta|)}$ and $\Sigma_3^{-1} = \mathrm{Diag}(|\alpha\cup\beta|, -I_{|\alpha\cup\beta|}) \in \mathbb{R}^{(1+|\alpha\cup\beta|)\times(1+|\alpha\cup\beta|)}$. In particular, the part $(\delta\sigma^{-1}\Sigma_3^{-1} + V_3V_3^\top)^{-1}V_3$ can be computed explicitly. Indeed, because $P_{\alpha\cup\beta}P^\top_{\alpha\cup\beta} = I_{|\alpha\cup\beta|}$ and $\delta = \sigma + \tau\sigma^{-1}$, one may obtain that

$$\frac{\delta}{\sigma}\Sigma_3^{-1} + V_3V_3^\top = \begin{pmatrix} \left(2+\frac{\tau}{\sigma^2}\right)|\alpha\cup\beta| & \boldsymbol{e}^\top_{|\alpha\cup\beta|} \\ \boldsymbol{e}_{|\alpha\cup\beta|} & -\frac{\tau}{\sigma^2}I_{|\alpha\cup\beta|}\end{pmatrix},$$

which is a symmetric quasidefinite matrix. Then, by the structure of V_3, one has

$$(\delta\sigma^{-1}\Sigma_3^{-1} + V_3V_3^\top)^{-1}V_3 = \begin{pmatrix} \frac{\sigma^2}{(\tau+\sigma^2)|\alpha\cup\beta|}\boldsymbol{e}^\top_{|\alpha\cup\beta|}P_{\alpha\cup\beta} \\ \frac{\sigma^2}{\tau}\left(\frac{\sigma^2}{(\tau+\sigma^2)|\alpha\cup\beta|}\boldsymbol{e}_{|\alpha\cup\beta|}\boldsymbol{e}^\top_{|\alpha\cup\beta|}P_{\alpha\cup\beta} - P_{\alpha\cup\beta}\right)\end{pmatrix}.$$

The cost of $A^\top_{\mathcal{J}}D^{-1}A_{\mathcal{J}}$ is $O(s^2(n + |\alpha\cup\beta| + 1))$.

Subcase 3.2: $|\alpha\cup\beta| > \theta_2|\gamma|$. We have that

$$D^{-1} = \frac{\sigma}{\tau}I_n - \frac{\sigma\varpi}{\tau}\left[\frac{1}{|\alpha\cup\beta|}\left(\boldsymbol{e}^\top_{|\alpha\cup\beta|}P_{\alpha\cup\beta}\right)^\top\left(\boldsymbol{e}^\top_{|\alpha\cup\beta|}P_{\alpha\cup\beta}\right) + P_\gamma^\top P_\gamma\right].$$

The cost of $A^\top_{\mathcal{J}}D^{-1}A_{\mathcal{J}}$ is $O(s^2(n + |\gamma| + 1))$.

Case 4: $\boldsymbol{u} \neq \overline{\boldsymbol{u}}, \|\overline{\boldsymbol{u}}\|_\infty = r, \|\overline{\boldsymbol{u}}\|_1 = kr, \beta \neq \emptyset$. It follows that

$$W = P^\top\mathrm{Diag}\left(I_{|\alpha|}, \frac{1}{|\beta|}E_{|\beta|\times|\beta|}, I_{|\gamma|}\right)P = P_\alpha^\top P_\alpha + \frac{1}{|\beta|}\left(\boldsymbol{e}^\top_{|\beta|}P_\beta\right)^\top\left(\boldsymbol{e}^\top_{|\beta|}P_\beta\right) + P_\gamma^\top P_\gamma.$$

Therefore,

$$D = \sigma[P_\alpha^\top P_\alpha + |\beta|^{-1}(\boldsymbol{e}^\top_{|\beta|}P_\beta)^\top(\boldsymbol{e}^\top_{|\beta|}P_\beta) + P_\gamma^\top P_\gamma] + \tau\sigma^{-1}I_n.$$

Subcase 4.1: $|\beta| < \theta_2(|\alpha| + |\gamma|)$. One obtains that

$$D^{-1} = \delta^{-1}[I_n - V_4^\top(\delta\sigma^{-1}\Sigma_4^{-1} + V_4V_4^\top)^{-1}V_4],$$

where $V_4^\top := (P_\beta^\top, (\boldsymbol{e}^\top_{|\beta|}P_\beta)^\top) \in \mathbb{R}^{n\times(|\beta|+1)}$ and $\Sigma_4^{-1} := \mathrm{Diag}(-I_{|\beta|}, |\beta|) \in \mathbb{R}^{(1+|\beta|)\times(1+|\beta|)}$. Similarly as in subcase 3.1, by making use of the structure of Σ_4^{-1} and V_4, one also gets the explicit expression $(\delta\sigma^{-1}\Sigma_4^{-1} + V_4V_4^\top)^{-1}V_4$ as follows:

$$\left(\frac{\delta}{\sigma}\Sigma_4^{-1} + V_4V_4^\top\right)^{-1}V_4 = \begin{pmatrix} \frac{\sigma^2}{\tau}\left(\frac{\sigma^2}{(\tau+\sigma^2)|\beta|}\boldsymbol{e}_{|\beta|}\boldsymbol{e}^\top_{|\beta|}P_\beta - P_\beta\right) \\ \frac{\sigma^2}{(\tau+\sigma^2)|\beta|}\boldsymbol{e}^\top_{|\beta|}P_\beta\end{pmatrix}.$$

Using this formula, the computational cost of $A^\top_{\mathcal{J}}D^{-1}A_{\mathcal{J}}$ is $O(s^2(n + |\beta| + 1))$.

Downloaded from informs.org by [158.132.13.163] on 10 April 2025, at 01:09 . For personal use only, all rights reserved.

Subcase 4.2: $|\beta| \geq \theta_2(|\alpha| + |\gamma|)$. We can obtain that

$$D^{-1} = \frac{\sigma}{\tau} I_n - \frac{\sigma\varpi}{\tau}\left[P_\alpha^\top P_\alpha + \frac{1}{|\beta|}\left(\boldsymbol{e}_{|\beta|}^\top P_\beta\right)^\top \left(\boldsymbol{e}_{|\beta|}^\top P_\beta\right) + P_\gamma^\top P_\gamma\right].$$

The computational cost of $A_{\mathcal{J}}^\top D^{-1} A_{\mathcal{J}}$ is then $O(s^2(n + |\alpha| + |\gamma| + 1))$.

Case 5. $\boldsymbol{u} \neq \overline{\boldsymbol{u}}, \|\overline{\boldsymbol{u}}\|_\infty = r, \|\overline{\boldsymbol{u}}\|_1 = kr, \beta = \emptyset$. Then, $M = 0_{n\times n}$, that is, $W = I_n$. Hence, $D = \delta I_n$, which implies $D^{-1} = \delta^{-1} I_n$, and the cost of $A_{\mathcal{J}}^\top D^{-1} A_{\mathcal{J}}$ reduces to $O(s^2 n)$.

These formulas indicate that, in computing the matrix multiplications $A_{\mathcal{J}}^\top D^{-1} A_{\mathcal{J}}$ as part of T, we only need $O(s^2 n)$ computational costs that are linear in n instead of $O(n^2 s + s^2 n)$ (recall that $s < \theta_1 n$ here). The inverse of $s \times s$ matrix T^{-1} can be then computed via the Cholesky factorization at $O(s^3)$ cost. Finally, we list the main computational cost of the semismooth Newton method for one iteration in Table 1.

4. Fast Computation of a Solution Path

The previous section focuses on solving (6) for a fixed value of $\lambda > 0$. One may also need to find a solution path of $\{\boldsymbol{x}(\lambda_i)\}_{i=1,\dots,N}$ for a given sequence of grid points $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_N \geq 0$ in order to select the best hyperparameter λ via the cross-validation. In this section, we present an adaptive sieving (AS) strategy to effectively reduce problem dimension in an iterative way so that a solution path can be quickly generated.

The AS strategy can be viewed as a special warm-start scheme that aggressively guesses and adjusts the support of $\boldsymbol{x}(\lambda_i)$ at a new grid point λ_i based on the solution $\boldsymbol{x}(\lambda_{i-1})$ computed at the previous grid point. Specifically, we start the ith problem in a restricted space by setting all components not in the support of $\boldsymbol{x}(\lambda_{i-1})$ as zero, which usually has a much smaller size than those given by the screening rules in El Ghaoui et al. (2012), Shibagaki et al. (2016), Ndiaye et al. (2015), and Wang et al. (2014). After solving the restricted problem by Algorithm 1, we update the support guided by the KKT conditions of the computed primal–dual pair and repeat this process until all KKT conditions of the original problem are satisfied.

The approach is motivated by a recent paper (Lin et al. 2020) on the computation of an exclusive Lasso solution path, in which the overall objective is the sum of a smooth function and a nonsmooth one. Because the CVaR function is not smooth, the method proposed in Lin et al. (2020) cannot be directly applied here. Our main contribution is to show that it suffices to consider only partial KKT conditions to guess and adjust the active features when Algorithm 1 is used as the subroutine to solve each reduced problem. The support is selected based on the following principle, which in fact is a direct consequence of Ndiaye et al. (2021, proposition 1). All proofs in this section are deferred to Online Appendix B.

Lemma 1. *Let* $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{z}}, \hat{\boldsymbol{u}})$ *be a KKT pair of Problem* (9). *Given* $j \in [d]$, *if* $\boldsymbol{x}^*$ *satisfies* $-(A^\top \hat{\boldsymbol{u}})_j \in \operatorname{int}(\partial(\lambda h)(\boldsymbol{x}^*))_j$, *then* $x_j^* = \hat{x}_j$.

Let $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{z}}, \hat{\boldsymbol{u}})$ be a KKT pair of Problem (9) with $\lambda = \lambda_i$. The preceding lemma indicates that only those elements in $\mathcal{J} := \{j \in [d] \,|\, -(A^\top \hat{\boldsymbol{u}})_j \notin \operatorname{int}(\partial(\lambda h)(\boldsymbol{x}^*))_j\}$ can be active, where $\boldsymbol{x}^*$ is a solution of the restricted problem in the space of the support. Because the dual solution $\hat{\boldsymbol{u}}$ is unknown, we take the expansion of the dual solution from the reduced

Table 1. The Main Computational Costs of Algorithm 2 in Each Iteration Under Solving Linear Equation (14) by the Direct Method

Step		Main terms	Cost	Total cost
Step 1	$s \geq \theta_1 n$	$A_{\mathcal{J}} A_{\mathcal{J}}^\top$	$O(n^2 s)$	$O(n^2(n+s))$
		Cholesky factorization of $\sigma A_{\mathcal{J}} A_{\mathcal{J}}^\top + \sigma W + \tau\sigma^{-1} I_n$	$O(n^3)$	
	$s < \theta_1 n$	$A_{\mathcal{J}}^\top D^{-1} A_{\mathcal{J}}$	$O(s^2 n)$	$O(s^2(n+s))$
		D^{-1}rhs	$O(n)$	
		Cholesky factorization of T	$O(s^3)$	
Step 2	$\nabla\varphi_t(\dot{\boldsymbol{u}})$	$A^\top \dot{\boldsymbol{u}}$	$O(nd)$	$O(n(d + \log n))$
		$\mathrm{Prox}_{\sigma\lambda\|\cdot\|_1}(\omega_1(\dot{\boldsymbol{u}}))$	$O(d)$	
		$A\mathrm{Prox}_{\sigma\lambda\|\cdot\|_1}(\omega_1(\dot{\boldsymbol{u}}))$	$O(nd)$	
		$\mathrm{Prox}_{\sigma\|\cdot\|_{(k)}}(\omega_2(\dot{\boldsymbol{u}}))$	$O(n \log n)$	
	$\varphi_t(\dot{\boldsymbol{u}})$	$A^\top \dot{\boldsymbol{u}}$	$O(nd)$	$O(n(d + \log n))$
		$\frac{1}{2}\|\omega_1(\dot{\boldsymbol{u}})\|^2 - e_{\sigma\lambda\|\cdot\|_1}(\omega_1(\dot{\boldsymbol{u}})) = \frac{1}{2}\|\mathrm{Prox}_{\sigma\lambda\|\cdot\|_1}(\omega_1(\dot{\boldsymbol{u}}))\|^2$	$O(d)$	
		$\frac{1}{2}\|\omega_2(\dot{\boldsymbol{u}})\|^2 - e_{\sigma\|\cdot\|_{(k)}}(\omega_2(\dot{\boldsymbol{u}})) = \frac{1}{2}\|\mathrm{Prox}_{\sigma\|\cdot\|_{(k)}}(\omega_2(\dot{\boldsymbol{u}}))\|^2$	$O(n \log n)$	

Downloaded from informs.org by [158.132.13.163] on 10 April 2025, at 01:09 . For personal use only, all rights reserved.

problem as its estimate. Denote the proximal residual function

$$R_\lambda(\boldsymbol{x},\boldsymbol{u}) := \boldsymbol{x} - \mathrm{Prox}_{\lambda h}(\boldsymbol{x} - A^\top \boldsymbol{u}). \tag{18}$$

We are now ready to present the detail of the AS strategy in Algorithm 3.

Algorithm 3 (Compute a Solution Path of Problem (9) for $\{\lambda_i\}_{i=1}^N$)

```
Choose an initial point $\bar{x}(\lambda_0) \in \mathbb{R}^d$ as the solution of (9) for $\lambda = \lambda_0$, a sequence of grid points $\lambda_0 > \lambda_1 > \cdots > \lambda_N > 0$, and a tolerance $\varepsilon \geq 0$. Compute the initial nonactive set $\mathcal{I}^*(\lambda_0) := \{1 \leq j \leq d \mid \bar{x}_j(\lambda_0) = 0\}$. Execute the following steps for $i = 1, 2, \ldots, N$ with the initialization $\mathcal{I}^0(\lambda_i) = \mathcal{I}^*(\lambda_{i-1})$ and $\nu = 0$.
1: Step 1. Find a KKT pair $(x^\nu(\lambda_i), z^\nu(\lambda_i), u^\nu(\lambda_i))$ of the following problem:
       $\underset{x\in\mathbb{R}^d, z\in\mathbb{R}^n}{\text{minimize}} \{f(z) + \lambda_i h(x) - \langle \delta_x, x\rangle - \langle \delta_z, z\rangle \mid Ax - z - b = \delta_c, x_{\mathcal{I}^\nu(\lambda_i)} = 0\}$,   (19)
   where $\delta_x \in \mathbb{R}^d$, $\delta_z \in \mathbb{R}^n$ and $\delta_c \in \mathbb{R}^n$ are error vectors satisfying
       $\max(\|\delta_x\|, \|\delta_z\|, \|\delta_c\|) \leq \frac{\varepsilon}{\sqrt{3}}$ and $(\delta_x)_{\mathcal{I}^\nu(\lambda_i)} = 0$.   (20)
   Compute $R_{\lambda_i}(x^\nu(\lambda_i), u^\nu(\lambda_i))$ defined in (18).
2: Step 2. If $\|R_{\lambda_i}(x^\nu(\lambda_i), u^\nu(\lambda_i))\| \leq \varepsilon$, stop and set $(x^*(\lambda_i), u^*(\lambda_i)) = (x^\nu(\lambda_i), u^\nu(\lambda_i))$ with $\mathcal{I}^*(\lambda_i) = \mathcal{I}^\nu(\lambda_i)$. Let $i \leftarrow i+1$ (unless $i = N$ already so that the algorithm should be stopped); otherwise, compute the index set
       $\mathcal{J}^{\nu+1}(\lambda_i) = \left\{ j \in \mathcal{I}^\nu(\lambda_i) \,\middle|\, -\left(A^\top u^\nu(\lambda_i)\right)_j \notin \lambda_i(\partial h(x^\nu(\lambda_i)))_j + \left(\frac{\varepsilon}{\sqrt{2|\mathcal{I}^\nu(\lambda_i)|}}\mathbb{B}_\infty\right)_j \right\}$
   and set $\mathcal{I}^{\nu+1}(\lambda_i) \leftarrow \mathcal{I}^\nu(\lambda_i) \setminus \mathcal{J}^{\nu+1}(\lambda_i)$. Set $\nu \leftarrow \nu + 1$ and go back to step 1.
```

Similarly as in Lin et al. (2020, proposition 2), we can establish the following finite convergence of Algorithm 3.

Proposition 1. *For each λ_i, Algorithm 3 terminates after a finite number of iterations with the output $(\boldsymbol{x}^*(\lambda_i), \boldsymbol{u}^*(\lambda_i))$ satisfying $\|R_{\lambda_i}(\boldsymbol{x}^*(\lambda_i), \boldsymbol{u}^*(\lambda_i))\| \leq \varepsilon$.*

The main computational burden in Algorithm 3 is to solve the constrained problems (19) repeatedly. Denote $\mathcal{I}^c := [d] \setminus \mathcal{I}$ and let $A_\mathcal{I} \in \mathbb{R}^{n\times|\mathcal{I}|}$ be the submatrix of A consisting of the columns of A indexed by $\mathcal{I}$. In fact, the tuple $(\bar{\boldsymbol{x}}, \bar{\boldsymbol{z}})$ together with the corresponding multiplier for the equality constraint can be obtained by approximately solving the following problem

$$\underset{\boldsymbol{x}\in\mathbb{R}^{|\mathcal{I}^c|}, \boldsymbol{z}\in\mathbb{R}^n}{\text{minimize}} \{f(\boldsymbol{z}) + \lambda_i h(\boldsymbol{x}) \mid A_{\mathcal{I}^c}\boldsymbol{x} - \boldsymbol{z} - \boldsymbol{b} = 0\}. \tag{21}$$

Let $(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{z}}, \tilde{\boldsymbol{u}})$ be an approximate KKT solution of the problem. Denote

$$(\boldsymbol{\delta_x})_{\mathcal{I}^c} = \tilde{\boldsymbol{x}} - \bar{\boldsymbol{x}}_{\mathcal{I}^c}, \quad \boldsymbol{\delta_z} = \tilde{\boldsymbol{z}} - \bar{\boldsymbol{z}}, \quad \boldsymbol{\delta_c} = A_{\mathcal{I}^c}\bar{\boldsymbol{x}}_{\mathcal{I}^c} - \bar{\boldsymbol{z}} - \boldsymbol{b},$$

where $\bar{\boldsymbol{x}}_{\mathcal{I}^c} := \mathrm{Prox}_{\lambda_i h}(\tilde{\boldsymbol{x}} - A_{\mathcal{I}^c}^\top \tilde{\boldsymbol{u}})$ and $\bar{\boldsymbol{z}} := \mathrm{Prox}_f(\tilde{\boldsymbol{u}} + \tilde{\boldsymbol{z}})$. Then, as long as $\max(\|(\boldsymbol{\delta_x})_{\mathcal{I}^c}\|, \|\boldsymbol{\delta_z}\|, \|\boldsymbol{\delta_c}\|) \leq \varepsilon/\sqrt{3}$, we have that Conditions (20) are satisfied by the pair $(\bar{\boldsymbol{x}}, \bar{\boldsymbol{z}})$, where $\bar{\boldsymbol{x}} \in \mathbb{R}^d$ is obtained by expanding $\bar{\boldsymbol{x}}_{\mathcal{I}^c}$ to a d-dimensional vector with the rest of the entries being 0.

Notice that we only use partial KKT conditions of Problem (9) as the stopping criterion of the AS strategy. In the following theorem, we show that the output tuple $(\bar{\boldsymbol{x}}, \bar{\boldsymbol{z}}, \bar{\boldsymbol{u}})$ of Algorithm 3 is indeed an approximate KKT solution of Problem (9) measured by all KKT conditions of Problem (9) provided that $\bar{\boldsymbol{x}} = \mathrm{Prox}_{\lambda_i h}(\bar{\boldsymbol{x}} + \boldsymbol{\delta_x} - A^\top \bar{\boldsymbol{u}})$.

Theorem 3. *Consider a fixed $\lambda_i > 0$ and a given tolerance $\varepsilon \geq 0$. Let $(\bar{\boldsymbol{x}}, \bar{\boldsymbol{z}}, \bar{\boldsymbol{u}})$ be a KKT solution of the reduced Problem (19) satisfying Conditions (20) and $\bar{\boldsymbol{x}}_\mathcal{I} = 0$. If $\bar{\boldsymbol{x}}$ further satisfies that $\bar{\boldsymbol{x}} = \mathrm{Prox}_{\lambda_i h}(\bar{\boldsymbol{x}} + \boldsymbol{\delta_x} - A^\top \bar{\boldsymbol{u}})$, then $(\bar{\boldsymbol{x}}, \bar{\boldsymbol{z}}, \bar{\boldsymbol{u}})$ is a KKT solution of the original problem:*

$$\underset{\boldsymbol{x}\in\mathbb{R}^d, \boldsymbol{z}\in\mathbb{R}^n}{\text{minimize}} \{f(\boldsymbol{z}) + \lambda_i h(\boldsymbol{x}) - \langle \boldsymbol{\delta_x}, \boldsymbol{x}\rangle - \langle \boldsymbol{\delta_z}, \boldsymbol{z}\rangle \mid A\boldsymbol{x} - \boldsymbol{z} - \boldsymbol{b} = \boldsymbol{\delta_c}\} \tag{22}$$

with $(\boldsymbol{\delta_x}, \boldsymbol{\delta_z}, \boldsymbol{\delta_c})$ satisfying Conditions (20).

It is worth mentioning that the condition $\bar{\boldsymbol{x}} = \mathrm{Prox}_{\lambda_i h}(\bar{\boldsymbol{x}} + \boldsymbol{\delta_x} - A^\top \bar{\boldsymbol{u}})$ in Theorem 3 is mild because we always have $\bar{\boldsymbol{x}} \approx \mathrm{Prox}_{\lambda_i h}(\bar{\boldsymbol{x}} + \boldsymbol{\delta_x} - A^\top \bar{\boldsymbol{u}})$ as long as the inequalities $\|R_{\lambda_i}(\bar{\boldsymbol{x}}, \bar{\boldsymbol{u}})\| \leq \varepsilon$ and $\|\boldsymbol{\delta_x}\| \leq \varepsilon/\sqrt{3}$ hold for sufficiently small

Downloaded from informs.org by [158.132.13.163] on 10 April 2025, at 01:09 . For personal use only, all rights reserved.

tolerance ε. Indeed, from the definition of $R_{\lambda_i}(\overline{\boldsymbol{x}},\overline{\boldsymbol{u}})$ in (18) and the Lipschitz continuity of $\mathrm{Prox}_{\lambda_i h}(\cdot)$, one may get that

$$\begin{aligned}\|\overline{\boldsymbol{x}} - \mathrm{Prox}_{\lambda_i h}(\overline{\boldsymbol{x}} + \boldsymbol{\delta}_{\boldsymbol{x}} - A^\top \overline{\boldsymbol{u}})\| &= \|R_{\lambda_i}(\overline{\boldsymbol{x}},\overline{\boldsymbol{u}}) + \mathrm{Prox}_{\lambda_i h}(\overline{\boldsymbol{x}} - A^\top \overline{\boldsymbol{u}}) - \mathrm{Prox}_{\lambda_i h}(\overline{\boldsymbol{x}} + \boldsymbol{\delta}_{\boldsymbol{x}} - A^\top \overline{\boldsymbol{u}})\| \\ &\le \|R_{\lambda_i}(\overline{\boldsymbol{x}},\overline{\boldsymbol{u}})\| + \|\boldsymbol{\delta}_{\boldsymbol{x}}\| \le \varepsilon + \varepsilon/\sqrt{3}.\end{aligned}$$

As a final remark, Problem (21) has the same form as (9) under a subset of measurement $A_{\mathcal{I}^c}$, which can be solved efficiently via the algorithm discussed in Section 3.

5. Extension to the VaR-Based Learning

Different from the CVaR that is convex in the random variable, the value at risk or, more generally, the truncated CVaR with $k_2 \neq 0$ in (7) are nonconvex risk measures. In this section, we apply an MM algorithm (Ortega and Rheinboldt 2000, section 8.3(d)) to solve the nonconvex problem (7), in which each convex subproblem is solved by a similar method as in the previous sections.

Consider the following problem:

$$\underset{\boldsymbol{x}\in\mathbb{R}^d}{\text{minimize}} \quad F(\boldsymbol{x}) := f_1(A\boldsymbol{x} - \boldsymbol{b}) - f_2(A\boldsymbol{x} - \boldsymbol{b}) + \lambda\|\boldsymbol{x}\|_1, \tag{23}$$

where $f_1 : \mathbb{R}^n \to (-\infty, +\infty]$ and $f_2 : \mathbb{R}^n \to \mathbb{R}$ are two convex functions. When $f_1 = \|\bullet\|_{(k_1)}, f_2 = \|\bullet\|_{(k_2)}$ and parameters $n \ge k_1 > k_2 > 0$, the problem reduces to the truncated CVaR problem in (7) with the absolute value loss. It is easy to see that the preceding objective function can be decomposed into the difference of two convex functions $f_1(A\boldsymbol{x} - \boldsymbol{b}) + \lambda\|\boldsymbol{x}\|_1$ and $f_2(A\boldsymbol{x} - \boldsymbol{b})$. Given an initial point $\boldsymbol{x}^0$, the MM algorithm consists of the following iterations:

$$\begin{aligned}\boldsymbol{x}^{\nu+1} \approx \underset{\boldsymbol{x}\in\mathbb{R}^d}{\operatorname{argmin}} \Big\{ &\hat{F}(\boldsymbol{x}; \rho_\nu, c_\nu, \boldsymbol{x}^\nu, \boldsymbol{a}^\nu) := f_1(A\boldsymbol{x} - \boldsymbol{b}) + \lambda\|\boldsymbol{x}\|_1 - f_2(A\boldsymbol{x}^\nu - \boldsymbol{b}) - (\boldsymbol{a}^\nu)^\top(A\boldsymbol{x} - A\boldsymbol{x}^\nu) \\ &+ \frac{\rho_\nu}{2}\|\boldsymbol{x} - \boldsymbol{x}^\nu\|^2 + \frac{c_\nu}{2}\|A\boldsymbol{x} - A\boldsymbol{x}^\nu\|^2 \Big\},\end{aligned} \tag{24}$$

where $\boldsymbol{a}^\nu \in \partial f_2(A\boldsymbol{x}^\nu - \boldsymbol{b})$ is a subgradient of f_2 at $A\boldsymbol{x}^\nu - \boldsymbol{b}$, and $\{\rho_\nu\}$ and $\{c_\nu\}$ are two positive convergent sequences on $\mathbb{R}$. Let $\boldsymbol{e}^{\nu+1}$ be a residual vector of the dual problem of (24) satisfying

$$\boldsymbol{u}^{\nu+1} \in \underset{\boldsymbol{u}\in\mathbb{R}^n}{\operatorname{argmin}}\{\phi(\boldsymbol{u}; \rho_\nu, c_\nu, \boldsymbol{x}^\nu, \boldsymbol{a}^\nu) - \langle \boldsymbol{u}, \boldsymbol{e}^{\nu+1}\rangle\}, \tag{25}$$

where ϕ is the corresponding negative dual objective function given by

$$\begin{aligned}\phi(\boldsymbol{u}; \rho_\nu, c_\nu, \boldsymbol{x}^\nu, \boldsymbol{a}^\nu) := c_\nu &\left[\frac{1}{2}\|c_\nu^{-1}\boldsymbol{u} + c_\nu^{-1}\boldsymbol{a}^\nu + A\boldsymbol{x}^\nu - \boldsymbol{b}\|^2 - e_{c_\nu^{-1}f_1}(c_\nu^{-1}\boldsymbol{u} + c_\nu^{-1}\boldsymbol{a}^\nu + A\boldsymbol{x}^\nu - \boldsymbol{b})\right] \\ &+ \rho_\nu\left[\frac{1}{2}\|\boldsymbol{x}^\nu - \rho_\nu^{-1}A^\top\boldsymbol{u}\|^2 - e_{\lambda\rho_\nu^{-1}\|\cdot\|_1}(\boldsymbol{x}^\nu - \rho_\nu^{-1}A^\top\boldsymbol{u})\right] \\ &+ \langle \boldsymbol{u}, \boldsymbol{b}\rangle - \frac{\rho_\nu}{2}\|\boldsymbol{x}^\nu\|^2 - \frac{c_\nu}{2}\|A\boldsymbol{x}^\nu - \boldsymbol{b}\|^2 - \langle \boldsymbol{a}^\nu, A\boldsymbol{x}^\nu - \boldsymbol{b}\rangle + f_2(A\boldsymbol{x}^\nu - \boldsymbol{b}).\end{aligned}$$

Then, we can build the convergence of the MM algorithm whose proof is presented in Online Appendix C. Recall that, for the difference-of-convex program $\text{minimize}_{\boldsymbol{x}}\,\{g(\boldsymbol{x}) - h(\boldsymbol{x})\}$, where $g : \mathbb{R}^n \to (-\infty, +\infty]$ and $\boldsymbol{h} : \mathbb{R}^n \to \mathbb{R}$ are two convex functions, we say a point $\boldsymbol{x}^* \in \mathbb{R}^n$ is a critical point if $\partial g(\boldsymbol{x}^*) \cap \partial h(\boldsymbol{x}^*) \neq \emptyset$; see, for example, Tao and An (1997).

Theorem 4. *Let $\{\boldsymbol{x}^\nu\}$ be a sequence generated by the MM algorithm. Assume that $\{\rho_\nu\}$ and $\{c_\nu\}$ are positive convergent sequences with $\lim_{\nu\to\infty}\rho_\nu > 0$, and the inequality*

$$2f_1(\boldsymbol{e}^{\nu+1}) + \frac{c_\nu}{2}\|\boldsymbol{e}^{\nu+1}\|^2 \le \frac{c_\nu}{8}\|A(\boldsymbol{x}^{\nu+1} - \boldsymbol{x}^\nu)\|^2 \tag{26}$$

holds for any $\nu \ge 0$, where $\boldsymbol{e}^{\nu+1}$ is given in (25). Then, (a) $F(\boldsymbol{x}^\nu) \ge F(\boldsymbol{x}^{\nu+1}) + (\rho_\nu/2)\|\boldsymbol{x}^{\nu+1} - \boldsymbol{x}^\nu\|^2$; (b) every cluster point $\boldsymbol{x}^\infty$ of the sequence $\{\boldsymbol{x}^\nu\}$ generated by the MM algorithm is a critical point of (23).

The subgradient of the nonseparable k-norm function can be formulated according to Watson (1992), Overton and Womersley (1993), and Wu et al. (2014). Notice that this subproblem takes almost the same form as the CVaR-based

Downloaded from informs.org by [158.132.13.163] on 10 April 2025, at 01:09 . For personal use only, all rights reserved.

regression (6) except for the two proximal terms. Similar to the MM framework in Tang et al. (2020), one can easily extend the properties and techniques in Algorithms 1 and 2 to efficiently solve such subproblems by the semismooth Newton method based on the proximal point algorithm. Details of the derivations are omitted.

6. Numerical Experiments

We have conducted extensive numerical experiments to demonstrate the effectiveness of the risk-sensitive models and the efficiency of our proposed algorithms. All the algorithms are implemented in MATLAB on a windows workstation (16-core, Intel(R) Core(TM) i7-10700 @ 2.90 GHz, 32 G RAM). The codes are available at the GitHub repository (Wu et al. 2022).

Throughout the experiments, we take the loss ℓ as the absolute value function.

6.1. Implementation Details

For the convex CVaR-based problem (9), we measure the qualities of the computed solutions in the following way:

$$\eta_{\text{res}} := \max\{\eta_{\text{p}}, \eta_{\text{d}}, \eta_{\text{gap}}\}, \tag{27}$$

where η_{p}, η_{d} and η_{gap} represent the relative primal infeasibility, the relative dual infeasibility, and the relative duality gap, respectively:

$$\begin{cases} \eta_{\text{p}} := \dfrac{\|A\boldsymbol{x} - \boldsymbol{z} - \boldsymbol{b}\|}{1 + \|\boldsymbol{b}\|}, \quad \eta_{\text{d}} := \max\left\{ \dfrac{\|A^\top \boldsymbol{u} - \Pi_{\mathcal{B}_\infty^\lambda}(A^\top \boldsymbol{u})\|}{1 + \|A^\top \boldsymbol{u}\|}, \dfrac{\|\boldsymbol{u} - \Pi_{\mathcal{B}^1_{(k)^*}}(\boldsymbol{u})\|}{1 + \|\boldsymbol{u}\|} \right\}, \\ \eta_{\text{gap}} := \dfrac{|\|A\boldsymbol{x} - \boldsymbol{b}\|_{(k)} + \lambda\|\boldsymbol{x}\|_1 + \langle \boldsymbol{u}, \boldsymbol{b}\rangle|}{1 + \|A\boldsymbol{x} - \boldsymbol{b}\|_{(k)} + \lambda\|\boldsymbol{x}\|_1 + |\langle \boldsymbol{u}, \boldsymbol{b}\rangle|}. \end{cases}$$

We also compute the following relative KKT residuals of the iterates

$$\eta_{\text{kkt}} = \max\{\eta_x, \eta_z, \eta_u\}, \tag{28}$$

where

$$\eta_x = \frac{\|A^\top \boldsymbol{u} + \Pi_{\mathcal{B}_\infty^\lambda}(\boldsymbol{x} - A^\top \boldsymbol{u})\|}{1 + \|\boldsymbol{x}\| + \|A^\top \boldsymbol{u}\|}, \quad \eta_z = \frac{\|\boldsymbol{u} - \Pi_{\mathcal{B}^1_{(k)^*}}(\boldsymbol{z} + \boldsymbol{u})\|}{1 + \|\boldsymbol{z}\| + \|\boldsymbol{u}\|}, \quad \eta_{\boldsymbol{u}} = \frac{\|A\boldsymbol{x} - \boldsymbol{z} - \boldsymbol{b}\|}{1 + \|\boldsymbol{b}\|}.$$

In addition, when two algorithms are stopped under different criteria, we use the objective values obtained from Gurobi with the tolerance 10^{-9} (denoted as $\text{obj}_{\text{Gurobi}}$) as benchmarks to test the quality of the computed objective values by both algorithms (denoted as $\text{obj}_{\text{computed}}$):

$$\text{relobj} := \frac{|\text{obj}_{\text{computed}} - \text{obj}_{\text{Gurobi}}|}{1 + |\text{obj}_{\text{Gurobi}}|}.$$

Given a tolerance $\varepsilon > 0$, we stop the algorithm if $\eta_{\text{res}} \le \varepsilon$ unless otherwise stated.

For the nonconvex VaR or truncated CVaR-based linear regressions, we stop the MM algorithm if

$$\text{obj-gap} := \frac{|F(\boldsymbol{x}^{t+1}) - F(\boldsymbol{x}^t)|}{1 + |F(\boldsymbol{x}^t)|} \le \varepsilon, \tag{29}$$

where F is the objective function defined in (23).

We set the maximum iteration numbers of both the outer proximal augmented Lagrangian and the inner semismooth Newton method to be 200. The parameters σ_t and τ_t in Algorithm 1 are updated according to the following rule:

$$\sigma_{t+1} = \begin{cases} \sigma_t & \text{if } \operatorname{mod}(t,3) \neq 0 \\ \min\{1.3\sigma_t, 10^6\} & \text{otherwise} \end{cases}; \quad \tau_{t+1} = \begin{cases} \tau_t & \text{if } \operatorname{mod}(t,3) \neq 0 \\ \max\{5\tau_t/6, 10^{-6}\} & \text{otherwise.} \end{cases}$$

Both synthetic data and the real data from the University of California, Irvine (UCI) data repository are used for the experiments.[4] In order to get sufficiently large test instances, we adopt the same manner as in Huang et al. (2010) to expand the original features of the UCI data using polynomial basis functions and append a digit number to the

Downloaded from informs.org by [158.132.13.163] on 10 April 2025, at 01:09 . For personal use only, all rights reserved.

data set name to indicate the degree of the used polynomial. We only present the results on the UCI data to save space. Results on the synthetic data are presented in Online Appendix F.

6.2. Solving the CVaR-Based Linear Regression for a Fixed λ

In this section, we compare the performance of our Algorithm 1 (N-ALM) with four other approaches under different accuracy: (a) the first order alternating direction method of multipliers (ADMM) and the smoothing method based on the inexact regularized proximal Newton method (S-IRPN) to get relatively low-accuracy solutions (with $\varepsilon = 10^{-3}$); (b) the Portfolio Safeguard (PSG) solver to get moderate-accuracy solutions (with $\varepsilon = 10^{-6}$); and (c) the barrier method in Gurobi (version 9.5.2) to get high-accuracy solutions (with $\varepsilon = 10^{-8}$). The detailed description of the ADMM and the S-IRPN to solve (9) are provided in Online Appendices D and E. We use the default settings in PSG and Gurobi for their respective termination. The ADMM is stopped if the total number of iterations reaches 30,000. The maximum time allowed for each method is two hours.

Table 2 shows the results of N-ALM, S-IRPN, and ADMM when $k = \lceil 0.1n \rceil$, $\lceil 0.5n \rceil$ and $\lceil 0.9n \rceil$, where all three algorithms are terminated by relobj $\le 10^{-3}$ because the S-IRPN does not return dual variables to compute η_{res} defined in (27). The column under "nnz" (here and in the rest of the tables) represents the number of nonzero entries of the solution $\boldsymbol{x}$ estimated in the way nnz $:= \min\{k \mid \|\boldsymbol{x}\|_{(k)} \ge 0.999\|\boldsymbol{x}\|_1\}$. Under the "iteration numbers" column of the N-ALM, we report the iteration counts for the outer loop proximal augmented Lagrangian method first, followed by the number of the inner semismooth Newton iterations in the bracket. As for the S-IRPN, we record the number of the approximating problems, followed by the number of iterations of the inner inexact regularized proximal Newton and the innermost coordinate descent algorithm in the bracket.

One may observe from Table 2 that the N-ALM can solve all the UCI instances within 92 seconds, but the ADMM and S-IRPN cannot solve the large instances loglp.E2006.train and triazines4 within two hours. For the small

Table 2. Results of the N-ALM, S-IRPN, and ADMM on UCI Data with $\varepsilon = 10^{-3}(\lambda := k\lambda_c\|A^\top \boldsymbol{b}\|_\infty)$

Proname				Relobj			Iteration numbers			Time (seconds)		
$n;d$	λ_c	k	nnz	N-ALM	S-IRPN	ADMM	N-ALM	S-IRPN	ADMM	N-ALM	S-IRPN	ADMM
triazines4	1e-4	19	891	9.8e-4	7.0e-3	1.0e-3	32(150)	3(83,39,510)	17,990	6.8	7,239.7	1,953.8
186;5,57,845		93	979	9.8e-4	1.0e-2	5.0e-3	35(150)	3(89,40,720)	30,000	6.4	7,447.5	3,217.7
		168	641	9.0e-4	5.7e-3	2.0e-3	41(151)	3(67,41,330)	30,000	6.4	7,546.2	3,229.9
pyrim5	1e-5	8	292	6.8e-4	6.2e-4	1.0e-3	16(49)	5(101,47,560)	7,319	0.6	995.3	113.2
74;1,69,911		37	209	9.7e-4	7.0e-3	1.0e-3	26(75)	4(221,347,860)	25,423	0.8	7,222.6	392.7
		67	192	8.5e-4	8.1e-4	1.0e-3	30(82)	4(119,176,020)	28,930	0.9	3,644.6	447.9
log1p.E2006.test	1e-6	331	73	7.0e-4	9.3e-4	1.0e-3	17(94)	2(27,5,590)	12,547	12.5	462.7	1,469.5
3,308;1,771,946		1,654	18	6.3e-4	2.2e-4	9.9e-4	11(40)	2(15,3,130)	18,516	4.0	256.6	2,145.4
		2,978	12	5.7e-4	3.3e-4	1.0e-3	12(33)	2(10,350)	22,495	3.0	31.8	2,607.3
bodyfat7	1e-7	26	256	8.8e-4	2.8e-2	1.0e-3	23(113)	9(337,180,800)	5,529	1.6	7,209.6	154.2
252;1,16,280		126	235	8.3e-4	2.9e-3	1.0e-3	36(168)	5(122,212,920)	13,039	2.2	7,295.7	356.7
		227	234	7.8e-4	9.7e-4	1.0e-3	40(186)	5(121,181,200)	13,769	2.6	6,134.1	373.8
housing7	1e-7	51	428	1.0e-3	7.0e-3	1.0e-3	32(137)	2(254,190,540)	9,857	3.2	7,223.7	386.9
506;77,520		253	456	8.8e-4	1.5e-3	1.0e-3	25(111)	6(267,194,270)	5,493	2.4	7,218.4	195.5
		456	411	8.5e-4	9.8e-4	1.0e-3	25(93)	3(201,135,340)	6,695	1.8	5,038.5	234.1
log1p.E2006.train	1e-7	1,609	124	7.5e-4	8.7e-4	1.8e-3	12(82)	2(22,5,760)	11,671	44.1	1,763.2	7,200.5
16,087;4,265,669		8,044	69	4.2e-4	6.1e-4	5.6e-3	8(42)	2(20,4,570)	11,688	14.4	1,399.2	7,200.3
		14,479	54	9.0e-4	6.2e-4	9.9e-3	9(37)	2(19,3,300)	11,704	10.8	1,012.7	7,200.2
mpg7	1e-7	40	224	7.9e-4	1.2e-3	9.9e-4	25(103)	9(480,410,710)	896	0.7	1,199.2	1.6
392;3,432		196	205	1.0e-3	9.5e-4	9.9e-4	21(85)	2(91,42,080)	487	0.4	122.7	0.7
		353	188	1.0e-3	1.0e-3	9.8e-4	19(67)	3(127,69,530)	773	0.2	195.1	1.0
abalone7	1e-8	418	157	9.9e-4	1.3e-1	9.9e-4	5(58)	4(708,109,890)	100	2.0	7,215.3	5.6
4,177;6,435		2,089	108	8.6e-4	7.4e-4	8.8e-4	3(28)	1(167,41,790)	120	0.8	2,212.4	5.8
		3,760	80	9.5e-4	8.7e-4	9.1e-4	5(43)	1(48,27,280)	135	0.6	1,428.3	6.4
space_ga9	1e-8	311	261	9.6e-4	9.8e-4	9.9e-4	28(672)	4(235,202,370)	1,285	14.4	4,766.9	37.9
3,107;5,005		1,554	223	9.2e-4	9.9e-4	9.2e-4	22(439)	3(116,116,120)	535	7.6	2,668.3	14.2
		2,797	214	9.4e-4	8.5e-4	9.8e-4	21(262)	3(97,77,690)	467	3.4	1,770.8	12.2
E2006.test	1e-9	331	325	8.8e-4	9.9e-4	1.0e-3	24(774)	4(111,11,100)	2,120	13.2	136.1	56.1
3,308;72,812		1,654	65	7.0e-4	7.7e-4	9.9e-4	8(265)	2(57,1,280)	1,666	3.1	17.8	45.5
		2,978	22	9.5e-4	6.5e-4	9.8e-4	5(29)	2(27,370)	1,514	0.3	5.7	41.6
E2006.train	1e-10	1,609	529	8.7e-4	8.5e-4	9.9e-4	10(972)	3(180,10,140)	2,073	91.7	599.6	765.6
16,087;1,50,348		8,044	105	6.6e-4	9.7e-4	1.0e-3	5(241)	1(46,1,030)	1,429	15.6	61.5	579.0
		14,479	52	9.1e-4	1.4e-4	1.0e-3	6(51)	2(46,1,090)	1,236	2.7	64.1	527.6

Downloaded from informs.org by [158.132.13.163] on 10 April 2025, at 01:09 . For personal use only, all rights reserved.

Table 3. Results of the N-ALM and the PSG Solver VAN on UCI Data with $\varepsilon = 10^{-6}$ ($\lambda := k\lambda_c \|A^\top \boldsymbol{b}\|_\infty$)

Probname	λ_c	k	solver	nnz	relobj	Objective values	Time (seconds)
mpg7	1e-7	40	N-ALM	162	6.46e-9	142.2669	0.9
392;3,432			VAN	178	6.07e-5	142.2756	7,200.4
		196	N-ALM	173	1.52e-7	447.1833	0.7
			VAN	181	2.04e-5	447.1923	6,037.5
		353	N-ALM	165	2.78e-9	537.2936	0.4
			VAN	171	3.25e-5	537.3111	7,202.2
abalone7	1e-8	418	N-ALM	102	4.97e-10	1,951.951	3.8
4,177;6,435			VAN	116	2.00e-5	1,951.990	4,383.6
		2,089	N-ALM	88	1.12e-8	5,108.126	2.7
			VAN	95	1.53e-5	5,108.204	2,920.8
		3,760	N-ALM	75	9.10e-9	6,099.619	2.8
			VAN	77	1.45e-5	6,099.708	3,091.5
space_ga9	1e-8	311	N-ALM	197	6.61e-9	61.86195	17.6
3,107;5,005			VAN	218	3.51e-5	61.86416	7,201.7
		1,554	N-ALM	178	4.27e-9	178.2515	13.5
			VAN	200	1.70e-5	178.2546	4,275.5
		2,797	N-ALM	160	1.36e-8	218.2444	14.0
			VAN	175	1.72e-5	218.2482	3,790.7

instances mpg7, space_ga9, and abalone7, the performance of the ADMM is comparable to the N-ALM and at least 126 times faster than the S-IRPN. However, for the large-scale instance loglp.E2006.test, the ADMM is much slower than the S-IRPN, and the latter is at least 10 times slower than the N-ALM. These results indicate that the N-ALM outperforms the S-IRPN and the ADMM even under low accuracy on the large-scale UCI data. It is also worth pointing out that the computational time of the N-ALM on most of the test instances decreases as the value of k increases. Further results of the N-ALM for different values of k can be found in Online Appendix F.

In Table 3, we evaluate the performance of our N-ALM to solve Problem (9) with seven solvers in the PSG package: VAN, TANK, CAR, BULDOZER, VANGRB, CARGRB, and HELI. These solvers use different optimization techniques to solve (9) with the latter three relying on Gurobi LP solvers to solve the subproblems (the detailed methods for each solver are not disclosed in the online manual of PSG). During our experiments, we found that VAN performs best among the seven solvers of PSG for solving Problem (9). Because the variable size of VAN cannot exceed 10,000, we only reported the results of our algorithm and VAN on three small-scale instances abalone7, mpg7, and space_gap9. Table 3 shows that our N-ALM is much more efficient and stable than VAN. For example, for the instances mpg7 with $k = 40$ and $k = 353$, VAN fails to obtain a good solution within two hours, whereas our N-ALM successfully solve them within one second.

In Table 4, we compare the performance of the N-ALM and the barrier method for the LP reformulation of (9) implemented by Gurobi. One can find that, on average, the N-ALM is about 26 times faster than Gurobi although the latter is comparable to the N-ALM for the smallest instance mpg7. For the largest instance log1p.E2006.train with $n = 16{,}087$ and $d = 4{,}265{,}669$, our approach is about 53 times faster than Gurobi.

6.3. Computing a Solution Path of the CVaR-Based Linear Regression

In this part, we demonstrate the effectiveness of the AS strategy combined with N-ALM (AS+N-ALM) for the ℓ_1-penalized CVaR-based problem (9) with a sequence of $\{\lambda_i\}_{i=1,\ldots,N}$. In order to guarantee the finite termination of Algorithm 3, we follow Proposition 1 to stop the N-ALM when $\eta_{\text{kkt}} \le 10^{-6}$, where η_{kkt} is the relative KKT residual defined in (28). We choose an initial index set $\mathcal{I}^*(\lambda_0)$ as follows (Lin et al. 2020):

$$\mathcal{I}^*(\lambda_0) = \{i \in [d] \,|\, c_i \text{ is among the first } \lceil \sqrt{d} \rceil \text{ largest values in } c_1, \ldots, c_d\},$$

where $c_i := |\langle a_i, \boldsymbol{b}\rangle| / \|a_i\|\|\boldsymbol{b}\|$ and a_i is the ith column vector of design matrix A for $i = 1, \ldots, d$.

In order to test the effectiveness of the AS+N-ALM for the computation of a solution path, we compare it with two other methods: the warm-started N-ALM (Warm+N-ALM) that takes the solution from the previous λ as the initial point and the pure N-ALM. We take equally spaced grid points λ with the number (esgp) and the range given in the second and third columns of Table 5, respectively. An interesting observation can be made from this table: for a fixed interval of λ, say $\lambda \in [1.5, 4.5]$, the average time of both AS+N-ALM and Warm+N-ALM is shorter if finer grids are used as the solution from the previous iteration is closer to the next one. The performance of

Downloaded from informs.org by [158.132.13.163] on 10 April 2025, at 01:09 . For personal use only, all rights reserved.

Table 4. Results of the N-ALM and Gurobi on UCI Data with $\varepsilon = 10^{-8}$ ($\lambda := k\lambda_c \|A^{\top}\boldsymbol{b}\|_{\infty}$)

Proname				Objective values		Iteration numbers		Time (seconds)	
$n;d$	λ_c	k	nnz	N-ALM	Gurobi	N-ALM	Gurobi	N-ALM	Gurobi
triazines4	1e-4	19	1,026	2.06652976	2.06652974	89(371)	42	16.2	342.2
186;5,57,845		93	804	6.87454698	6.87454700	75(284)	59	11.7	464.2
		168	690	9.90365457	9.90365458	78(281)	55	11.5	445.0
pyrim5	1e-5	8	89	0.09927193	0.09927193	80(194)	25	2.1	24.3
74;1,69,911		37	89	0.12563268	0.12563269	80(199)	30	2.1	28.4
		67	89	0.15290241	0.15290242	80(195)	32	2.1	27.6
log1p.E2006.test	1e-6	331	75	291.872709	291.872709	62(227)	23	23.7	374.7
3,308;17,71,946		1,654	11	764.447381	764.447381	64(190)	22	15.6	288.7
		2,978	7	991.113836	991.113836	67(173)	14	13.8	285.8
bodyfat7	1e-7	26	233	0.00227986	0.00227986	60(221)	31	2.9	97.9
252;1,16,280		126	233	0.01104852	0.01104853	70(264)	40	3.4	114.6
		227	233	0.01990488	0.01990488	71(274)	47	3.5	118.6
housing7	1e-7	51	342	139.054736	139.054736	95(541)	28	9.5	146.0
506;77,520		253	355	456.549602	456.549600	111(631)	30	11.7	153.8
		456	326	597.894136	597.894136	73(233)	29	4.0	152.3
log1p.E2006.train	1e-7	1,609	146	1,356.80447	1,356.80447	66(326)	18	116.4	4,434.2
16,087;4,265,669		8,044	71	3,562.21589	3,562.22408	66(294)	15	79.6	7,204.4
		14,479	38	4,476.78391	4,476.78391	71(353)	20	85.2	2,714.1
mpg7	1e-7	40	162	142.266885	142.266885	72(212)	17	0.9	3.4
392;3,432		196	173	447.183182	447.183182	63(214)	15	0.7	3.2
		353	164	537.293571	537.293586	87(173)	18	0.5	3.5
abalone7	1e-8	418	102	1,951.95134	1,951.95134	63(217)	19	4.2	251.8
4,177;6,435		2,089	88	5,108.12623	5,108.12625	51(210)	18	3.1	241.0
		3,760	75	6,099.61915	6,099.61916	58(246)	17	3.3	234.6
space_ga9	1e-8	311	197	61.8619528	61.8619526	83(992)	17	18.9	356.4
3,107;5,005		1,554	178	178.251538	178.251539	84(1,015)	18	14.5	361.6
		2,797	160	218.244432	218.244432	77(1,311)	14	14.5	283.4
E2006.test	1e-9	331	317	246.275458	246.275458	62(936)	21	15.9	88.0
3,308;72,812		1,654	72	644.312599	644.312599	53(967)	23	11.9	43.6
		2,978	32	770.693080	770.693080	55(1,019)	11	11.9	51.1
E2006.train	1e-10	1,609	563	1,303.27104	1,303.27104	64(1,983)	20	162.7	1,960.6
16,087;1,50,348		8,044	126	3,433.28303	3,433.28304	67(2,235)	16	124.3	532.0
		14,479	60	4,175.14735	4,175.14735	69(2,665)	21	134.2	378.8

AS+N-ALM is consistently better than the Warm+N-ALM and N-ALM for all the test instances, which can also be seen clearly from Figure 1 on the cumulative time used to generate each path. In particular, the AS+N-ALM is about 2.5 and 5 times faster than Warm+N-ALM and N-ALM, respectively, for loglp.E2006.test and 2.2 and 4.2 times faster, respectively, for loglp.E2006.train. The superior performance of the AS strategy may be partially explained by Figure 2: the average reduced subproblem size (mean(n)) by this strategy matches the actual number of nonzeros in the optimal solution quite well.

6.4. Solving the Nonconvex Truncated CVaR-Based Linear Regression

The last part of the results is about the MM algorithm associated the semismooth Newton method based on the proximal point algorithm (MM+N-PPA) for the truncated CVaR-based problem (23), for which we compare its performance with the barrier method in Gurobi for solving the QP reformulation of Subproblem (24) (MM+Gurobi). Because the origin is a good initial point for the MM algorithm when the solution is sufficiently sparse, we compute a low-accuracy solution ($\varepsilon = 10^{-4}$) by the N-ALM for the convex problem (23) with $k_2 = 0$ as an initial point of MM only when a small λ_c is taken on each instance. The parameters k_1 and k_2 are chosen from $\{(k_1, k_2) | (\lceil 0.9n \rceil, \lceil 0.9n \rceil - 1), (\lceil 0.1n \rceil, \lceil 0.1n \rceil - 1), (\lceil 0.9n \rceil, \lceil 0.1n \rceil)\}$.

The results are summarized in Table 6. It can be observed that all the instances can be solved successfully by MM+N-PPA under the criterion obj-gap $< 10^{-6}$, where obj-gap is defined in (29). The symbol "-" indicates that MM+Gurobi reports failure for that instance as it is out of memory when computing the $d \times d$ matrix $A^{\top}A$ in the QP form of Subproblem (24). One can also find that our MM+N-PPA can be at least 289 times faster than the MM+Gurobi on the small-scale instance mpg7.

Downloaded from informs.org by [158.132.13.163] on 10 April 2025, at 01:09 . For personal use only, all rights reserved.

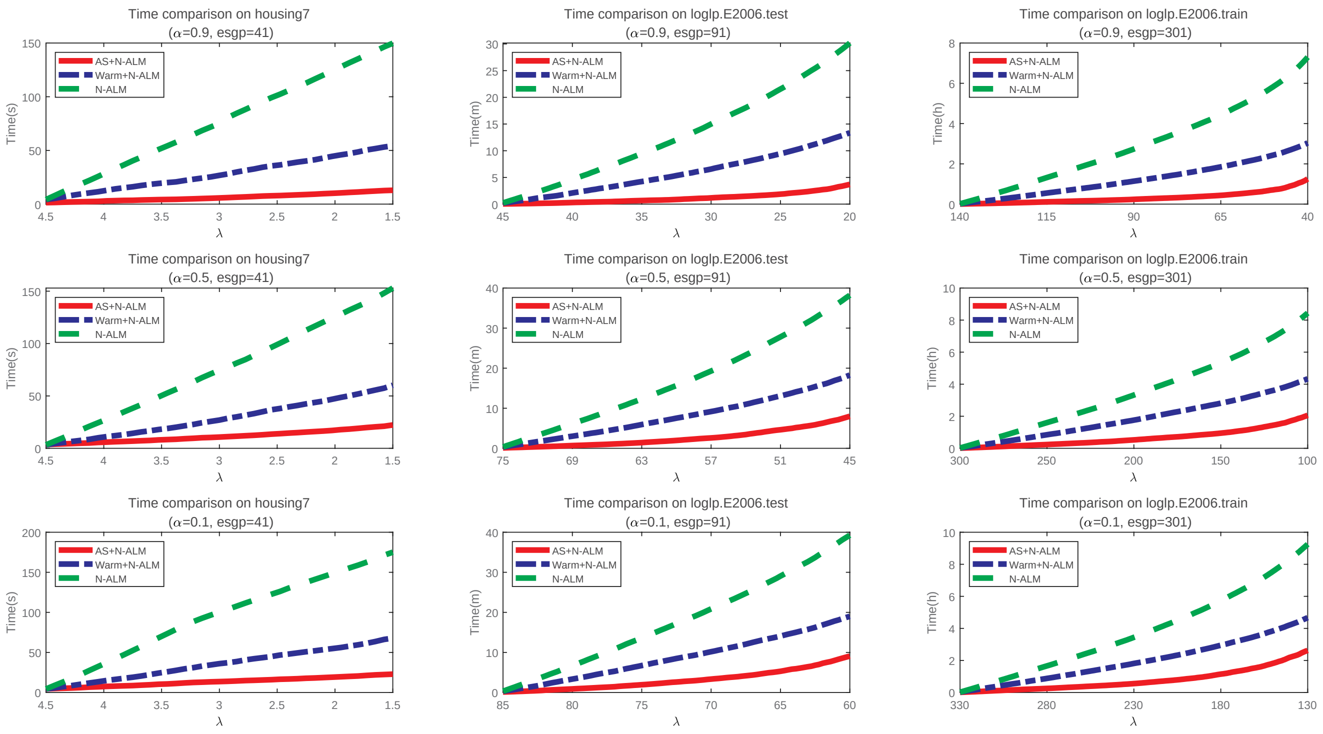
Table 5. The Average Time in Seconds (Total Time/esgp) Spent by Each Algorithm to Generate the Whole Solution Path

		$k=\lceil 0.1n\rceil$					$k=\lceil 0.5n\rceil$					$k=\lceil 0.9n\rceil$				
Probname $n;d$	esgp	range of λ	nnz	a	b	c	range of λ	nnz	a	b	c	range of λ	nnz	a	b	c
housing7 506;77,520	21	[4.5:-0.15:1.5]	106	0.40	1.57	3.63	[4.5:-0.15:1.5]	159	0.71	1.72	3.76	[4.5:-0.15:1.5]	178	0.71	1.92	4.26
	31	[4.5:-0.1:1.5]	106	0.34	1.41	3.60	[4.5:-0.1:1.5]	159	0.60	1.54	3.72	[4.5:-0.1:1.5]	178	0.62	1.76	4.27
	41	[4.5:-0.075:1.5]	106	0.31	1.32	3.65	[4.5:-0.075:1.5]	159	0.54	1.47	3.73	[4.5:-0.075:1.5]	178	0.56	1.65	4.27
loglp.E2006.test 3,308;1,771,946	31	[45:-5/6:20]	62	3.37	11.18	19.88	[75:-1:45]	126	7.08	15.41	25.21	[85:-5/6:60]	143	7.35	15.72	25.86
	61	[45:-5/12:20]	61	2.64	9.56	19.91	[75:-0.5:45]	126	5.82	13.35	25.13	[85:-5/12:60]	143	6.23	13.83	26.14
	91	[45:-5/18:20]	61	2.44	8.81	19.89	[75:-1/3:45]	126	5.27	12.05	25.19	[85:-5/18:60]	143	5.99	12.54	25.88
loglp.E2006.train 16,087;4,265,669	101	[140:-1:40]	81	18.48	49.26	88.87	[300:-2:100]	122	30.15	65.90	101.39	[330:-2:130]	147	36.92	73.60	111.83
	201	[140:-1/2:40]	81	15.85	40.80	87.83	[300:-1:100]	121	25.94	57.00	100.85	[330:-1:130]	147	32.99	62.40	110.84
	301	[140:-1/3:40]	81	14.99	36.38	87.30	[300:-2/3:100]	121	24.66	51.85	100.96	[330:-2/3:130]	147	31.44	55.88	110.66

Note. a := AS+N-ALM, b := Warm+N-ALM, c := N-ALM.

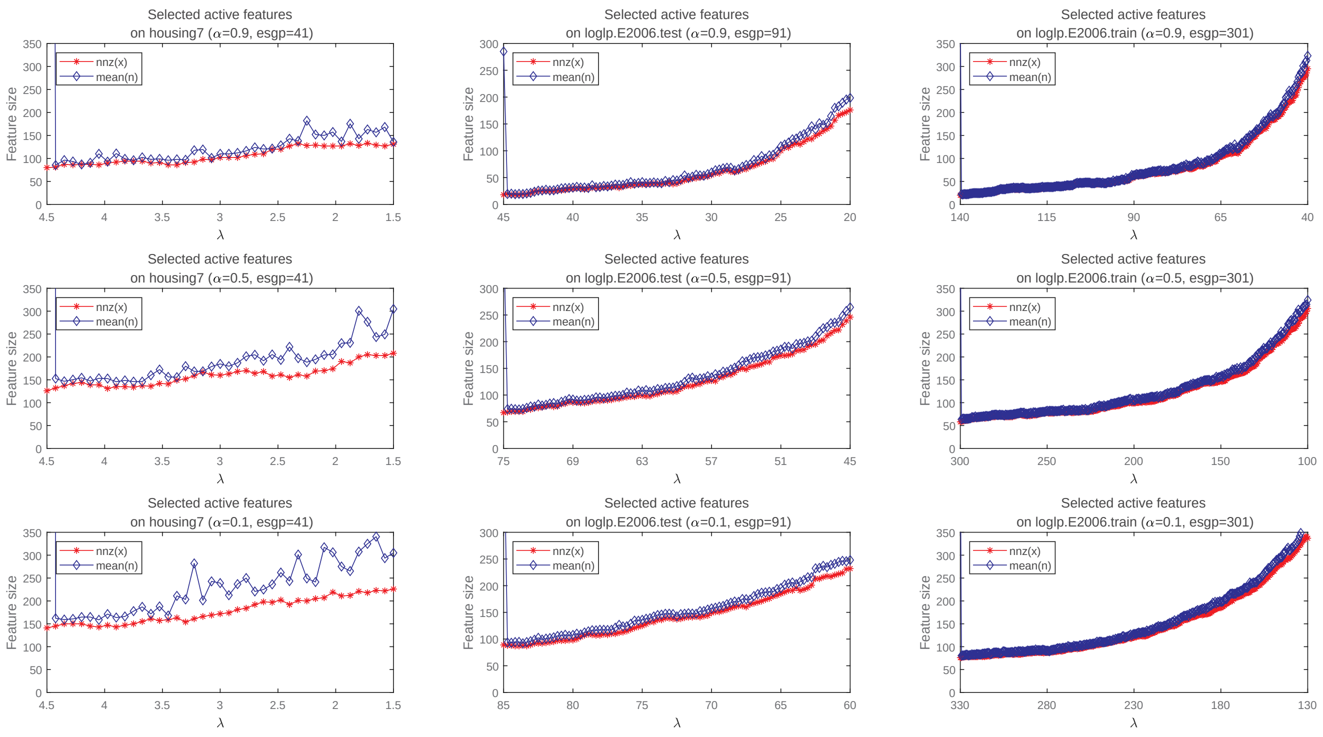
Downloaded from informs.org by [158.132.13.163] on 10 April 2025, at 01:09 . For personal use only, all rights reserved.

Figure 1. (Color online) The Cumulative Time Spent by Each Method to Generate the Whole Solution Path



Note. s := second, m := minute, h := hour.

Figure 2. (Color online) The Average Number of Selected Active Features by the AS Strategy (Diamond Line) vs. the True Cardinality of the Solution (Star Line)



Downloaded from informs.org by [158.132.13.163] on 10 April 2025, at 01:09 . For personal use only, all rights reserved.

Table 6. Results for Solving the Truncated CVaR-Based Linear Regression on UCI Data

Probname					Obj-gap		Objective values		MM iterations		Time (seconds)	
$n;d$	k_1	k_2	λ_c	nnz	a	b	a	b	a	b	a	b
mpg7 392;3,432	353	352	1e-5	70	5.4e-7	3.4e-7	2.83044	2.83040	65	96	0.9	5,487.5
			1e-6	56	8.5e-7	5.8e-7	0.77057	1.05063	60	39	1.3	2,691.7
	40	39	1e-5	29	6.5e-8	7.3e-7	8.01852	9.56507	111	98	2.2	6,175.7
			1e-6	87	4.3e-7	2.3e-7	3.72543	4.43914	46	50	1.5	3,930.6
	353	40	1e-5	10	7.0e-7	2.2e-8	1583.89	1583.89	22	2	0.8	1,152.0
			1e-6	50	7.4e-9	2.7e-7	546.890	546.890	7	6	1.4	401.4
space_ga9 3,107;5,005	2,797	2,796	1e-5	19	3.5e-7	1.3e-4	0.08696	0.33700	91	8	178.5	7,386.1
			1e-6	53	6.8e-7	5.3e-5	0.05408	0.06993	32	6	151.3	8,356.9
	311	310	1e-5	16	9.6e-7	1.2e-4	0.30552	0.81928	83	7	52.2	7,601.6
			1e-6	57	1.4e-7	5.2e-4	0.20351	0.25755	46	5	122.0	8,082.6
	2,797	311	1e-5	7	4.5e-8	1.2e-2	284.128	284.661	13	9	4.6	7,974.0
			1e-6	21	3.9e-10	2.9e-4	211.450	211.488	11	7	88.8	7,273.6
pyrim5 74;1,69,911	67	66	1e-5	102	9.4e-7	—	0.00037	—	30	—	3.3	—
			1e-6	131	7.8e-7	—	0.00005	—	12	—	1.5	—
	8	7	1e-5	138	1.2e-7	—	0.04651	—	30	—	5.0	—
			1e-6	308	9.2e-7	—	0.00009	—	6	—	1.6	—
	67	8	1e-5	101	8.4e-7	—	0.04889	—	28	—	9.0	—
			1e-6	88	3.6e-7	—	0.00483	—	8	—	2.5	—
bodyfat7 252;1,16,280	227	226	1e-5	24	1.0e-6	—	0.00302	—	35	—	23.9	—
			1e-6	37	5.9e-7	—	0.00034	—	17	—	26.9	—
	26	25	1e-5	101	6.5e-7	—	0.00441	—	23	—	27.3	—
			1e-6	258	6.3e-7	—	0.00082	—	7	—	6.3	—
	227	26	1e-5	30	3.8e-14	—	0.59856	—	15	—	16.2	—
			1e-6	173	9.1e-11	—	0.06235	—	6	—	41.8	—
housing7 506;77,520	456	455	1e-5	159	2.5e-9	—	4.03777	—	89	—	13.8	—
			1e-6	107	1.0e-9	—	0.74782	—	97	—	79.9	—
	51	50	1e-5	69	9.3e-7	—	11.5785	—	103	—	11.0	—
			1e-6	151	9.5e-7	—	5.03373	—	59	—	9.7	—
	456	51	1e-5	27	1.4e-7	—	2,410.42	—	20	—	4.3	—
			1e-6	145	1.8e-8	—	915.881	—	9	—	19.3	—

Note. a := MM+N-PPA, b := MM+Gurobi, $\lambda := (k_1 - k_2)\lambda_c\|A^\top \boldsymbol{b}\|_\infty$.

Acknowledgments
The authors thank anonymous reviewers and the associate editor for the valuable comments and suggestions that helped improve the paper and software greatly.

Endnotes
[1] See http://www.aorda.com/index.php/portfolio-safeguard/.

[2] The codes are available at http://www.dingchao.info/codes/.

[3] The conjugate gradient method is used in our experiment if one of the following cases is satisfied: (i) $n > 5{,}000$ and $s > 3{,}000$; (ii) $n > 2{,}000$ and $s > 8{,}000$; (iii) $n > 100$ and $s > 10{,}000$.

[4] See https://archive.ics.uci.edu/ml/index.php.

References
Alfons A, Croux C, Gelper S (2013) Sparse least trimmed squares regression for analyzing high-dimensional large data sets. *Ann. Appl. Statist.* 7(1):226–248.
Bengio Y, Louradour J, Collobert R, Weston J (2009) Curriculum learning. *Proc. 26th Annual Internat. Conf. Machine Learn.* (ACM, New York), 41–48.
Bertsekas D, Nedic A, Ozdaglar A (2003) *Convex Analysis and Optimization* (Athena Scientific, Belmont, MA).
Bhatia R (1997) *Matrix Analysis* (Springer-Verlag, New York).
Brucker P (1984) An $O(n)$ algorithm for quadratic knapsack problems. *Oper. Res. Lett.* 3(3):163–166.
Cui Y, Ding C, Li XD, Zhao XY (2021) Augmented Lagrangian methods for convex optimization problems. *J. Oper. Res. Soc. China* 10(2):305–342.
El Ghaoui L, Viallon V, Rabbani T (2012) Safe feature elimination for the lasso and sparse supervised learning problems. *Pacific J. Optim.* 8(4):667–698.
Facchinei F, Pang JS (2007) *Finite-Dimensional Variational Inequalities and Complementarity Problems* (Springer Science & Business Media, New York).

Downloaded from informs.org by [158.132.13.163] on 10 April 2025, at 01:09 . For personal use only, all rights reserved.

Fan J, Lv J (2008) Sure independence screening for ultrahigh dimensional feature space. *J. Roy. Statist. Soc. B* 70(5):849–911.

Fan K (1951) Maximum properties and inequalities for the eigenvalues of completely continuous operators. *Proc. Natl. Acad. Sci. USA* 37(11):760–766.

Hastie T, Tibshirani R, Wainwright M (2015) *Statistical Learning with Sparsity: The Lasso and Generalizations* (CRC Press, Boca Raton, FL).

Held M, Wolfe P, Crowder HP (1974) Validation of subgradient optimization. *Math. Programming* 6(1):62–88.

Helgason K, Kennington J, Lall H (1980) A polynomially bounded algorithm for a singly constrained quadratic program. *Math. Programming* 18(1):338–343.

Hiriart-Urruty JB, Strodiot JJ, Nguyen VH (1984) Generalized Hessian matrix and second-order optimality conditions for problems with $C^{1,1}$ data. *Appl. Math. Optim.* 11(1):43–56.

Horn RA, Johnson CR (2013) *Matrix Analysis*, 2nd ed. (Cambridge University Press, Cambridge, UK).

Huang L, Jia J, Yu B, Chun BG, Maniatis P, Naik M (2010) Predicting execution time of computer programs using sparse polynomial regression. Lafferty J, Williams C, Shawe-Taylor J, Zemel R, Culotta A, eds. *Adv. Neural Inform. Processing Systems*, vol. 23 (NeurIPS, San Diego), 883–891.

Levy D, Carmon Y, Duchi JC, Sidford A (2020) Large-scale methods for distributionally robust optimization. Larochelle H, Ranzato M, Hadsell R, Balcan MF, Lin H, eds. *Adv. Neural Inform. Processing Systems*, vol. 33 (NeurIPS, San Diego), 8847–8860.

Li XD, Sun DF, Toh KC (2018) On efficiently solving the subproblems of a level-set method for fused lasso problems. *SIAM J. Optim.* 28(2):1842–1866.

Li XD, Sun DF, Toh KC (2020) An asymptotically superlinearly convergent semismooth Newton augmented Lagrangian method for linear programming. *SIAM J. Optim.* 30(3):2410–2440.

Lin MX, Yuan YC, Sun DF, Toh KC (2020) Adaptive sieving with PPDNA: Generating solution paths of exclusive lasso models. Preprint, submitted September 18, https://arxiv.org/abs/2009.08719.

Lyu S, Fan Y, Ying Y, Hu BG (2020) Average top-k aggregate loss for supervised learning. *IEEE Trans. Pattern Anal. Machine Intelligence* 44(1):76–86.

Ndiaye E, Fercoq O, Salmon J (2021) Screening rules and its complexity for active set identification. *J. Convex Anal*. 28(4):1053–1072.

Ndiaye E, Fercoq O, Gramfort A, Salmon J (2015) Gap safe screening rules for sparse multi-task and multi-class models. Cortes C, Lawrence N, Lee D, Sugiyama M, Garnett R, eds. *Adv. Neural Inform. Processing Systems*, vol. 28 (NeurIPS, San Diego), 811–819.

Ortega JM, Rheinboldt WC (2000) *Iterative Solution of Nonlinear Equations in Several Variables*, Classics in Applied Mathematics, vol. 30 (SIAM, Philadelphia).

Overton ML, Womersley RS (1993) Optimality conditions and duality theory for minimizing sums of the largest eigenvalues of symmetric matrices. *Math. Programming* 62(1):321–357.

Pardalos PM, Kovoor N (1990) An algorithm for a singly constrained class of quadratic programs subject to upper and lower bounds. *Math. Programming* 46(1):321–328.

Rockafellar RT (1976) Augmented Lagrangians and applications of the proximal point algorithm in convex programming. *Math. Oper. Res.* 1(2):97–116.

Rockafellar RT, Uryasev S (2000) Optimization of conditional value-at-risk. *J. Risk* 2(3):21–41.

Rockafellar RT, Wets RJB (1998) *Variational Analysis* (Springer-Verlag, Berlin).

Rousseeuw PJ (1984) Least median of squares regression. *J. Amer. Statist. Assoc.* 79(388):871–880.

Shibagaki A, Karasuyama M, Hatano K, Takeuchi I (2016) Simultaneous safe screening of features and samples in doubly sparse modeling. Balcan MF, Weinberger KQ, eds. *Proc. 33rd Internat. Conf. Machine Learn.*, vol. 48 (ICML, San Diego), 1577–1586.

Shrivastava A, Gupta A, Girshick R (2016) Training region-based object detectors with online hard example mining. *Proc. IEEE Conf. Comput. Vision Pattern Recognition*, 761–769.

Sun J (1986) On monotropic piecewise quadratic programming. Unpublished PhD thesis, Department of Mathematics, University of Washington, Seattle.

Tang P, Wang C, Sun D, Toh KC (2020) A sparse semismooth Newton based proximal majorization-minimization algorithm for nonconvex square-root-loss regression problems. *J. Machine Learn. Res.* 21(226):1–38.

Tao PD, An LTH (1997) Convex analysis approach to DC programming: Theory, algorithms and applications. *Acta Mathematica Vietnamica* 22(1):289–355.

Tibshirani R, Bien J, Friedman J, Hastie T, Simon N, Taylor J, Tibshirani RJ (2012) Strong rules for discarding predictors in lasso-type problems. *J. Roy. Statist. Soc. B* 74(2):245–266.

Wang H, Li G, Jiang G (2007) Robust regression shrinkage and consistent variable selection through the LAD-lasso. *J. Bus. Econom. Statist.* 25(3):347–355.

Wang J, Zhou J, Liu J, Wonka P, Ye J (2014) A safe screening rule for sparse logistic regression. Ghahramani Z, Welling M, Cortes C, Lawrence N, Weinberger KQ, eds. *Adv. Neural Inform. Processing Systems*, vol. 27 (NeurIPS, San Diego), 1053–1061.

Watson GA (1992) Linear best approximation using a class of polyhedral norms. *Numer. Algorithms* 2(3):321–336.

Wu B, Ding C, Sun DF, Toh KC (2014) On the Moreau-Yosida regularization of the vector k-norm related functions. *SIAM J. Optim.* 24(2):766–794.

Wu C, Cui Y, Li D, Sun D (2022) Convex and nonconvex risk-based linear regression at scale. URL http://dx.doi.org/10.5281/zenodo.7483279, available at https://github.com/INFORMSJoC/2022.0012.

Xu H, Caramanis C, Mannor S (2010) Robust regression and lasso. *IEEE Trans. Inform. Theory* 56(7):3561–3574.

Zhao XY, Sun DF, Toh KC (2010) A Newton-CG augmented Lagrangian method for semidefinite programming. *SIAM J. Optim.* 20(4):1737–1765.