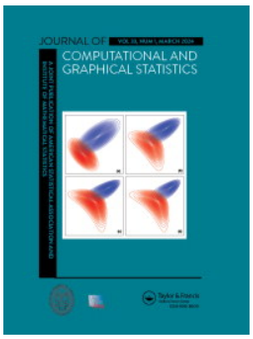


Journal of Computational and Graphical Statistics

ISSN: (Print) (Online) Journal homepage: www.tandfonline.com/journals/ucgs20

A Unified Algorithm for Penalized Convolution Smoothed Quantile Regression

Rebeka Man, Xiaoou Pan, Kean Ming Tan & Wen-Xin Zhou

To cite this article: Rebeka Man, Xiaoou Pan, Kean Ming Tan & Wen-Xin Zhou (26 Dec 2023): A Unified Algorithm for Penalized Convolution Smoothed Quantile Regression, Journal of Computational and Graphical Statistics, DOI: 10.1080/10618600.2023.2275999

To link to this article: https://doi.org/10.1080/10618600.2023.2275999

JOURNAL OF COMPUTATIONAL AND GRAPHICAL STATISTICS
2023, VOL. 00, NO. 0, 1–13
https://doi.org/10.1080/10618600.2023.2275999

A Unified Algorithm for Penalized Convolution Smoothed Quantile Regression

Rebeka Man[a], Xiaoou Pan[b], Kean Ming Tan[a], and Wen-Xin Zhou[c]

[a]Department of Statistics, University of Michigan, Ann Arbor, MI; [b]Department of Mathematics, University of California, San Diego, La Jolla, CA; [c]Department of Information and Decision Sciences, University of Illinois at Chicago, Chicago, IL

ABSTRACT

Penalized quantile regression (QR) is widely used for studying the relationship between a response variable and a set of predictors under data heterogeneity in high-dimensional settings. Compared to penalized least squares, scalable algorithms for fitting penalized QR are lacking due to the non-differentiable piecewise linear loss function. To overcome the lack of smoothness, a recently proposed convolution-type smoothed method brings an interesting tradeoff between statistical accuracy and computational efficiency for both standard and penalized quantile regressions. In this article, we propose a unified algorithm for fitting penalized convolution smoothed quantile regression with various commonly used convex penalties, accompanied by an R-language package `conquer` available from the Comprehensive R Archive Network. We perform extensive numerical studies to demonstrate the superior performance of the proposed algorithm over existing methods in both statistical and computational aspects. We further exemplify the proposed algorithm by fitting a fused lasso additive QR model on the world happiness data.

ARTICLE HISTORY
Received April 2022
Accepted October 2023

KEYWORDS
Convolution smoothing; Lasso; Majorize-minimization algorithm; Penalized optimization; Quantile estimation regression

1. Introduction

Let $y \in \mathbb{R}$ be a scalar response variable of interest, and $\boldsymbol{x} \in \mathbb{R}^p$ be a p-dimensional vector of covariates. Since the seminal work of Koenker and Bassett (1978), quantile regression (QR) has become an indispensable tool for understanding pathways of dependence between y and $\boldsymbol{x}$, which is irretrievable through conditional mean regression analysis via the least squares method. Motivated by a wide range of applications, various models have been proposed and studied for QR, from parametric to nonparametric and from low- to high-dimensional covariates. We refer the reader to Koenker (2005) and Koenker et al. (2017) for a comprehensive exposition of quantile regression.

Consider a high-dimensional linear QR model in which the number of covariates, p, is larger than the number of observations, n. In this setting, different low-dimensional structures have been imposed on the regression coefficients, thus, motivating the use of various penalty functions. One of the most widely used assumption is the sparsity, which assumes that only a small number of predictors are associated with the response. Quantile regression models are capable of capturing heterogeneity in the set of important predictors at different quantile levels of the response distribution caused by, for instance, heteroscedastic variance. For fitting sparse models in general, various sparsity-inducing penalties have been introduced, such as the lasso (ℓ_1-penalty) (Tibshirani 1996), elastic net (hybrid of ℓ_1/ℓ_2-penalty) (Zou and Hastie 2005), smoothly clipped absolute deviation (SCAD) penalty (Fan and Li 2001) and minimax concave (MC) penalty (Zhang 2010). Other commonly used regularizers that induce different types of structures include the group lasso (Yuan and Lin 2006), sparse group lasso (Simon et al. 2013) and fused lasso (Tibshirani et al. 2005), among others. We refer to the monographs Bühlmann and van de Geer (2011), Hastie, Tibshirani, and Wainwright (2015), and Wainwright (2019) for systematic introductions of high-dimensional statistical learning.

The above penalties/regularizers have been extensively studied when applied with the least squares method, accompanied with the user-friendly and efficient software `glmnet` (Friedman, Hastie, and Tibshirani 2010). Quantile regression, on the other hand, involves minimizing a non-differentiable piecewise linear loss, known as the check function. Although a handful of algorithms have been developed based on either linear programming or the alternating direction method of multipliers (ADMM) (Koenker and Ng 2005; Li and Zhu 2008; Peng and Wang 2015; Yu, Lin, and Wang 2017; Gu et al. 2018), there is not much software that is nearly as efficient as `glmnet` available for penalized quantile regressions. As the most recent progress, Yi and Huang (2017) proposed a semismooth Newton coordinate descent (SNCD) algorithm for solving a Huberized QR with the elastic net penalty; Yu, Lin, and Wang (2017) and Gu et al. (2018), respectively proposed ADMM-based algorithms for solving folded-concave penalized QR. See Gu et al. (2018) for a detailed computational comparison between the SNCD and ADMM-based algorithms, which favors the latter. The primary computation effort of each ADMM update is to evaluate the inverse of a $p \times p$ or $n \times n$ matrix. This can be computationally expensive when both n and p are large. Compared to sparsity-inducing regularization, the computational develop-

CONTACT Wen-Xin Zhou wenxinz@uic.edu Department of Information and Decision Sciences, University of Illinois at Chicago, Chicago, IL.

ment of penalized QR with other important penalties, such as the group lasso, is still scarce. For instance, quantile regression with the group lasso penalty can be formulated as a second-order cone programming (SOCP) problem (Kato 2011), solvable by general-purpose optimization toolboxes. These toolboxes are only adapted to small-scale problems and usually lead to solution with very high precision (low duality gap). For large-scale datasets, they tend to be too slow, or often run out of memory.

Addressing the non-differentiability of an objective function through smoothing has been a common technique in the statistics and machine learning literature. Several studies have proposed smoothing methods in the context of rank estimation, which bears many similarities to quantile regression. In their work, Brown and Wang (2005) introduced an induced smoothing method designed to smooth rank estimators, with a particular focus on estimating standard errors and covariance matrices. Building upon this method, Pang, Lu, and Wang (2012) employed it to develop a variance estimation procedure for the censored quantile coefficient estimator. They also used an algorithm based on linear programming. Additionally, Choi and Choi (2021) followed the convolution-type smoothing technique outlined in Fernandes, Guerre, and Horta (2021) to propose a smoothed estimating equation procedure. This procedure uses a logistic kernel approximation to smooth the estimating functions for the semiparametric accelerated failure time model, especially when dealing with high-dimensional right-censored data. Furthermore, they introduced a coordinate descent algorithm for the ℓ_1-penalized, adaptive ℓ_1-penalized, and SCAD-penalized least squares approximations of their smoothed objective function.

To resolve the non-differentiability issue of the check loss, Horowitz (1998) proposed to smooth the check loss directly using a kernel function. This approach, however, leads to a non-convex loss function which brings further computational issues especially in high dimensions. Recently, Fernandes, Guerre, and Horta (2021) employed a convolution-type smoothing technique to introduce the smoothed quantile regression (SQR) without sacrificing convexity. Convolution smoothing turns the non-differentiable check function into a twice-differentiable, convex and locally strongly convex surrogate, which admits fast and scalable gradient-based algorithms to perform optimization (He et al. 2023). Theoretically, the SQR estimator is asymptotically first-order equivalent to the standard QR estimator, and enjoys desirable statistical properties (Fernandes, Guerre, and Horta 2021; He et al. 2023). For high-dimensional sparse models, Tan, Wang, and Zhou (2022) proposed an iteratively reweighted ℓ_1-penalized SQR estimator that achieves oracle rate of convergence when the signals are sufficiently strong. They also proposed coordinate descent and ADMM-based algorithms for (weighted) ℓ_1-penalized SQR with the uniform and Gaussian kernels. These algorithms, however, do not adapt to more general kernel functions as well as penalties.

In this article, we introduce a major variant of the local adaptive majorize-minimization (LAMM) algorithm (Fan et al. 2018) for fitting convolution smoothed quantile regression that applies to any kernel function and a wide range of convex penalties. The main idea is to construct an isotropic quadratic objective function that locally majorizes the smoothed quantile loss such that closed-form updates are available at each iteration. The quadratic coefficient is adaptively chosen in order to guarantee the decrease of the objective function. In a sense, the LAMM algorithm can be viewed as a generalization of the iterative shrinkage-thresholding algorithm (ISTA) (Beck and Teboulle 2009). Compared to the interior point methods (for solving linear programming and SOCP problems) as well as ADMM, LAMM is a simpler gradient-based algorithm that is particularly suited for large-scale problems, where the dominant computational effort is a relatively cheap matrix-vector multiplication at each step. The (local) strong convexity of the convolution smoothed quantile loss facilitates the convergence of such a first order method. A key advantage of the proposed algorithm over those in Tan, Wang, and Zhou (2022) is that it can be applied to a broad class of convex penalties, typified by the lasso, elastic net, group lasso and sparse group lasso, and to any continuous kernel function. The proposed algorithm has been implemented in the R package `conquer` (He et al. 2022). This package provides a comprehensive framework for fitting penalized (smoothed) quantile regression models, encompassing all the penalties discussed in this article.

The remainder of the article is organized as follows. Section 2 briefly revisits quantile regression and its convolution smoothed counterpart. In Section 3, we describe a general local adaptive majorize-minimization principal for solving penalized smoothed quantile regression with four types of convex penalties, which are the lasso (ℓ_1-penalty), elastic net (a combination of ℓ_1- and ℓ_2-penalties), group lasso (weighted ℓ_2-penalty) and sparse group lasso (a combination of ℓ_1- and weighted ℓ_2-penalties). The computational and statistical efficiency of the proposed algorithm is demonstrated via extensive simulation studies in Section 4. In Section 5, we further exemplify the proposed algorithm by fitting a fused lasso additive QR model on the world happiness data.

2. Quantile Regression and Convolution Smoothing

Let $\boldsymbol{x} \in \mathbb{R}^p$ be a p-dimensional covariates and let $y \in \mathbb{R}$ be a scalar response variable. Given a quantile level $\tau \in (0,1)$ of interest, assume that the τth conditional quantile of y given $\boldsymbol{x}$ follows a linear model $F^{-1}_{y|\boldsymbol{x}}(\tau) = \boldsymbol{x}^T\boldsymbol{\beta}^*(\tau)$, where $\boldsymbol{\beta}^*(\tau) = \{\beta^*_1(\tau), \ldots, \beta^*_p(\tau)\}^T \in \mathbb{R}^p$. For notational convenience, we set $x_1 \equiv 1$ such that β^*_1 is the intercept. Moreover, we suppress the dependency of $\boldsymbol{\beta}^*(\tau)$ on τ throughout the article. Let $\{y_i, \boldsymbol{x}_i\}_{i=1}^n$ be a random sample of size n from $(y, \boldsymbol{x})$. The standard quantile regression estimator is defined as the solution to the optimization problem (Koenker and Bassett 1978)

$$\underset{\boldsymbol{\beta}\in\mathbb{R}^p}{\text{minimize}} \ \frac{1}{n}\sum_{i=1}^{n} \rho_\tau(y_i - \boldsymbol{x}_i^T\boldsymbol{\beta}), \tag{2.1}$$

where $\rho_\tau(u) = u\{\tau - \mathbb{1}(u < 0)\}$ is the quantile loss, also known as the check function. Although the quantile loss is convex, its non-differentiability (even only at one point) prevents gradient-based algorithms to be efficient. In this case, subgradient methods typically exhibit very slow (sublinear) convergence and hence are not computationally stable. A more widely acknowledged approach is to formulate the optimization problem (2.1) as a linear program, solvable by the simplex algorithm or interior

point methods. The latter has an average-case computational complexity that grows as a cubic function of p (Portnoy and Koenker 1997).

For fitting a sparse QR model in high dimensions, a natural parallel to the lasso (Tibshirani 1996) is the ℓ_1-penalized QR (QR-lasso) estimator (Belloni and Chernozhukov 2011), defined as a solution to the optimization problem

$$\underset{\boldsymbol{\beta}\in\mathbb{R}^p}{\text{minimize}} \ \frac{1}{n}\sum_{i=1}^{n}\rho_\tau(y_i-\boldsymbol{x}_i^T\boldsymbol{\beta})+\lambda\|\boldsymbol{\beta}\|_1, \tag{2.2}$$

where $\lambda > 0$ is a regularization parameter that controls (indirectly) the sparsity level of the solution, and $\|\cdot\|_1$ denotes the ℓ_1-norm. We refer to Wang, Wu, and Li (2012) and Zheng, Peng, and He (2015) for further extensions to adaptive ℓ_1 and folded concave penalties. Computationally, all of these sparsity-driven penalized methods boil down to solving a weighted ℓ_1-penalized QR loss minimization problem that is of the form

$$\underset{\boldsymbol{\beta}\in\mathbb{R}^p}{\text{minimize}} \ \frac{1}{n}\sum_{i=1}^{n}\rho_\tau(y_i-\boldsymbol{x}_i^T\boldsymbol{\beta})+\sum_{j=1}^{p}\lambda_j|\beta_j|, \tag{2.3}$$

where $\lambda_j \geq 0$ for each $1 \leq j \leq p$. Thus far, the most notable methods for solving (2.2) or more generally (2.3) include linear programming (Koenker 2023), coordinate descent algorithms (Peng and Wang 2015; Yi and Huang 2017) and ADMM-based algorithms (Yu, Lin, and Wang 2017; Gu et al. 2018). Among these, ADMM-based algorithms have the best overall performance as documented in Gu et al. (2018). The dominant computational effort of each ADMM update is the inversion of a $p\times p$ or an $n \times n$ matrix. For genomics data that typically has a small sample size, say in the order of hundreds, ADMM works well even when the dimension p is in the order of thousands or tens of thousands. However, there is not much efficient algorithm, that is also scalable to n, available for (weighted) ℓ_1-penalized QR, let alone for more general penalties.

To address the non-differentiability of the quantile loss, Fernandes, Guerre, and Horta (2021) proposed a convolution smoothed approach to quantile regression, resulting in a twice-differentiable, convex and (locally) strongly convex loss function. On the statistical aspect, Fernandes, Guerre, and Horta (2021) and He et al. (2023) established the asymptotic and finite-sample properties for the smoothed QR (SQR) estimator, respectively. Tan, Wang, and Zhou (2022) further considered the penalized SQR with iteratively reweighted ℓ_1-regularization and proved oracle properties under a minimum signal strength condition. Let $K(\cdot)$ be a symmetric, nonnegative kernel function, that is, $K(u) = K(-u) \geq 0$ for any $u \in \mathbb{R}$, and $\int_{-\infty}^{\infty}K(u)\mathrm{d}u = 1$. Given a bandwidth parameter $h > 0$, the ℓ_1-penalized SQR (SQR-lasso) estimator is defined as the solution to

$$\underset{\boldsymbol{\beta}\in\mathbb{R}^p}{\text{minimize}} \ \frac{1}{n}\sum_{i=1}^{n}\ell_{h,\tau}(y_i-\boldsymbol{x}_i^T\boldsymbol{\beta})+\lambda\|\boldsymbol{\beta}\|_1, \quad \text{where}$$

$$\ell_{h,\tau}(u)=\frac{1}{h}\int_{-\infty}^{\infty}\rho_\tau(v)K\left(\frac{v-u}{h}\right)\mathrm{d}v. \tag{2.4}$$

Equivalently, the smoothed loss $\ell_{h,\tau}$ can be written as $\ell_{h,\tau} = \rho_\tau \circ K_h$, where $K_h(u) = (1/h)K(u/h)$ and "$\circ$" denotes the convolution operator. Below, we present four most frequently used kernel functions, accompanied by the corresponding explicit expressions for the resultant smoothed check losses.

1. (Uniform kernel) The uniform kernel takes the form $K(u) = \frac{1}{2}\mathbb{1}(|u| \leq 1)$. For any $h > 0$ and $\tau \in (0, 1)$, the corresponding smoothed check loss is given by $\ell_{h,\tau}(u) = \tau u - u\mathbb{1}(u < -h) + \frac{1}{4h}(u-h)^2\mathbb{1}(|u| \leq h)$.
2. (Gaussian kernel) The Gaussian kernel takes the form $K(u) = (2\pi)^{-1/2}e^{-u^2/2}$. The corresponding smoothed loss is $\ell_{h,\tau}(u) = \{\tau - \Phi(-u/h)\}u + (2\pi)^{-1/2}he^{-u^2/(2h^2)}$.
3. (Logistic kernel) The logistic kernel takes the form $K(u) = e^{-u}/(1+e^{-u})^2$. The corresponding smoothed loss is $\ell_{h,\tau}(u) = \tau u + h\log(1+e^{-u/h})$.
4. (Laplacian kernel) The Laplacian kernel takes the form $K(u) = \frac{1}{2}e^{-|u|}$. In this case, the smoothed loss function is given by $\ell_{h,\tau}(u) = \ell_\tau(u) + 0.5he^{-|u|/h}$.

We refer to Remark 3.1 in He et al. (2023) for details on Epanechnikov and triangle kernels.

Statistical properties of convolution smoothed QR have been studied in the context of linear models under fixed-p, growing-p and high-dimensional settings (Fernandes, Guerre, and Horta 2021; Tan, Wang, and Zhou 2022; He et al. 2023). In the low-dimension setting "$p \ll n$", He et al. (2023) showed that the SQR estimator is (asymptotically) first-order equivalent to the QR estimator. Moreover, the asymptotic normality of SQR holds under a weaker requirement on dimensionality than needed for QR. In the high-dimensional regime "$p \gg n$", Tan, Wang, and Zhou (2022) proved that the SQR-lasso estimator with a properly chosen bandwidth enjoys the same convergence rate as QR-lasso (Belloni and Chernozhukov 2011). With iteratively reweighted ℓ_1-regularization, oracle properties can be achieved by SQR under a weaker signal strength condition than needed for QR.

For fitting ℓ_1-penalized SQR, Tan, Wang, and Zhou (2022) proposed coordinate descent and ADMM-based algorithms that are tailored to the uniform and Gaussian kernels, respectively. These algorithms do not exhibit evident advantages over existing ones for fitting penalized QR, and also limit the choice of both kernel and penalty functions. The main contribution this article is to develop a computationally efficient generic algorithm for penalized SQR, which applies to any nonnegative kernel smoothing function and various convex penalties. With the lasso penalty, the proposed algorithm is much more scalable than those in Tan, Wang, and Zhou (2022). The numerical studies in Section 4 further demonstrate the efficacy of the proposed algorithm for group lasso SQR.

3. Local Adaptive Majorize-minimization Algorithm for Penalized SQR

In this section, we describe a unified algorithm to solve the optimization problem for penalized SQR, which has a general form

$$\underset{\boldsymbol{\beta}\in\mathbb{R}^p}{\text{minimize}} \ \frac{1}{n}\sum_{i=1}^{n}\ell_{h,\tau}(y_i-\boldsymbol{x}_i^T\boldsymbol{\beta})+P(\boldsymbol{\beta}), \tag{3.1}$$

where $P(\boldsymbol{\beta})$ is a generic convex penalty function and $\ell_{h,\tau}(\cdot)$ is the smoothed check loss given in (2.4). In this article, we focus on the following four widely used convex penalty functions.

1. Weighted lasso (Tibshirani 1996): $P(\boldsymbol{\beta}) = \sum_{j=1}^{p} \lambda_j |\beta_j|$, where $\lambda_j \geq 0$ for $j = 1, \ldots, p$;
2. Elastic net (Zou and Hastie 2005): $P(\boldsymbol{\beta}) = \lambda\alpha\|\boldsymbol{\beta}\|_1 + \lambda(1-\alpha)\|\boldsymbol{\beta}\|_2^2$, where $\lambda > 0$ is a sparsity-inducing parameter and $\alpha \in [0,1]$ is a user-specified constant that controls the tradeoff between the ℓ_1 penalty and the ridge penalty;
3. Group lasso (Yuan and Lin 2006): $P(\boldsymbol{\beta}) = \lambda \sum_{g=1}^{G} w_g \|\boldsymbol{\beta}_g\|_2$, where $\boldsymbol{\beta} = (\boldsymbol{\beta}_1^T, \ldots, \boldsymbol{\beta}_G^T)^T$ and $\boldsymbol{\beta}_g$ is a sub-vector of $\boldsymbol{\beta}$ corresponding to the gth group of coefficients, and $w_g > 0$ are predetermined weights;
4. Sparse group lasso (Simon et al. 2013): $P(\boldsymbol{\beta}) = \lambda\|\boldsymbol{\beta}\|_1 + \lambda \sum_{g=1}^{G} w_g \|\boldsymbol{\beta}_g\|_2$.

We employ the local adaptive majorize-minimization (LAMM) principle to derive an iterative algorithm for solving (3.1). The LAMM principle is a generalization of the majorize-minimization (MM) algorithm (Lange, Hunter, and Yang 2000; Hunter and Lange 2004) to high dimensions, and has been applied to penalized least squares, generalized linear models (Fan et al. 2018) and robust regression (Pan, Sun, and Zhou 2021). We first provide a brief overview of the LAMM algorithm.

Consider the minimization of a general smooth function $f(\boldsymbol{\beta})$. Given an estimate $\widehat{\boldsymbol{\beta}}^{k-1}$ at the kth iteration, the LAMM algorithm locally majorizes $f(\boldsymbol{\beta})$ by a properly constructed function $g(\boldsymbol{\beta}|\widehat{\boldsymbol{\beta}}^{k-1})$ that satisfies the local property

$$f(\widehat{\boldsymbol{\beta}}^{k}) \leq g(\widehat{\boldsymbol{\beta}}^{k}|\widehat{\boldsymbol{\beta}}^{k-1}) \qquad \text{and} \qquad g(\widehat{\boldsymbol{\beta}}^{k-1}|\widehat{\boldsymbol{\beta}}^{k-1}) = f(\widehat{\boldsymbol{\beta}}^{k-1}), \tag{3.2}$$

where $\widehat{\boldsymbol{\beta}}^{k} = \operatorname{argmin}_{\boldsymbol{\beta}} g(\boldsymbol{\beta}|\widehat{\boldsymbol{\beta}}^{k-1})$. This ensures the decrease of the objective function after each step, that is, $f(\widehat{\boldsymbol{\beta}}^{k}) \leq f(\widehat{\boldsymbol{\beta}}^{k-1})$. Note that (3.2) is a relaxation of the global majorization requirement, $f(\boldsymbol{\beta}) \leq g(\boldsymbol{\beta}|\widehat{\boldsymbol{\beta}}^{k-1})$, used in the MM algorithm (Lange, Hunter, and Yang 2000; Hunter and Lange 2004).

Motivated by the local property in (3.2), we now derive an iterative algorithm for solving (3.1). For notational convenience, let $Q(\boldsymbol{\beta}) = n^{-1}\sum_{i=1}^{n} \ell_{h,\tau}(y_i - \boldsymbol{x}_i^T\boldsymbol{\beta})$ and let $\nabla Q(\boldsymbol{\beta})$ be the gradient of $Q(\boldsymbol{\beta})$. We locally majorize $Q(\boldsymbol{\beta})$ given $\widehat{\boldsymbol{\beta}}^{k-1}$ by constructing an isotropic quadratic function of the form

$$F(\boldsymbol{\beta}|\phi_k, \widehat{\boldsymbol{\beta}}^{k-1}) = Q(\widehat{\boldsymbol{\beta}}^{k-1}) + \langle \nabla Q(\widehat{\boldsymbol{\beta}}^{k-1}), \boldsymbol{\beta} - \widehat{\boldsymbol{\beta}}^{k-1} \rangle + \frac{\phi_k}{2}\|\boldsymbol{\beta} - \widehat{\boldsymbol{\beta}}^{k-1}\|_2^2,$$

where $\phi_k > 0$ is a quadratic parameter (to be determined) at the kth iteration. Then, define the kth iterate $\widehat{\boldsymbol{\beta}}^{k}$ as the solution to

$$\underset{\boldsymbol{\beta}\in\mathbb{R}^p}{\text{minimize}}\ F(\boldsymbol{\beta}|\phi_k, \widehat{\boldsymbol{\beta}}^{k-1}) + P(\boldsymbol{\beta}). \tag{3.3}$$

To ensure the descent of the objective function in (3.1) at each iteration, the parameter $\phi_k > 0$ needs to be sufficiently large such that $Q(\widehat{\boldsymbol{\beta}}^{k}) \leq F(\widehat{\boldsymbol{\beta}}^{k}|\phi_k, \widehat{\boldsymbol{\beta}}^{k-1})$. Consequently,

$$\begin{aligned} Q(\widehat{\boldsymbol{\beta}}^{k}) + P(\widehat{\boldsymbol{\beta}}^{k}) &\leq F(\widehat{\boldsymbol{\beta}}^{k}|\phi_k, \widehat{\boldsymbol{\beta}}^{k-1}) + P(\widehat{\boldsymbol{\beta}}^{k}) \\ &\leq F(\widehat{\boldsymbol{\beta}}^{k-1}|\phi_k, \widehat{\boldsymbol{\beta}}^{k-1}) + P(\widehat{\boldsymbol{\beta}}^{k-1}) \\ &= Q(\widehat{\boldsymbol{\beta}}^{k-1}) + P(\widehat{\boldsymbol{\beta}}^{k-1}), \end{aligned}$$

where the second inequality is due to the fact that $\widehat{\boldsymbol{\beta}}^{k}$ is a minimizer of (3.3). In practice, we choose ϕ_k by starting from a small value $\phi_0 = 0.01$ and successively inflate it by a factor $\gamma > 1$ until the majorization requirement $Q(\widehat{\boldsymbol{\beta}}^{k}) \leq F(\widehat{\boldsymbol{\beta}}^{k}|\phi_k, \widehat{\boldsymbol{\beta}}^{k-1})$ is met at each iteration of the LAMM algorithm. The tuning parameter γ serves a role similar to the control parameters used in the backtracking line search method. In the context of penalized least squares, Fan et al. (2018) recommended employing $\gamma = 2$. However, due to the increased condition number of the Hessian matrix for the smoothed empirical quantile loss compared to the squared loss, we opt for a smaller value of $\gamma = 1.2$ to enhance the accuracy of the quadratic approximation. Through extensive numerical studies, we see that this choice consistently yields favorable results across various settings.

One of the main advantages of our approach is that the isotropic form of $F(\boldsymbol{\beta}|\phi_k, \widehat{\boldsymbol{\beta}}^{k-1})$, as a function of $\boldsymbol{\beta}$, permits a simple analytic solution $\widehat{\boldsymbol{\beta}}^{k} = (\widehat{\beta}_1^k, \ldots, \widehat{\beta}_p^k)^T$ for different convex penalty functions $P(\boldsymbol{\beta})$. By the first-order optimization condition, $\widehat{\boldsymbol{\beta}}^{k}$ satisfies

$$\mathbf{0} \in \nabla_{\boldsymbol{\beta}} Q(\widehat{\boldsymbol{\beta}}^{k-1}) + \phi_k(\widehat{\boldsymbol{\beta}}^{k} - \widehat{\boldsymbol{\beta}}^{k-1}) + \partial P(\boldsymbol{\beta})|_{\boldsymbol{\beta}=\widehat{\boldsymbol{\beta}}^{k}},$$

where ∂P denotes the subdifferential of $P : \mathbb{R}^p \to [0, \infty)$. With certain convex penalties, a closed-form expression for $\widehat{\boldsymbol{\beta}}^{k}$ can be derived from the above condition. Since a common practice is to leave the intercept term unpenalized, its update takes a simple form $\widehat{\beta}_1^k = \widehat{\beta}_1^{k-1} - \phi_k^{-1}\nabla_{\beta_1} Q(\widehat{\boldsymbol{\beta}}^{k-1})$. We summarize the update rules of $\widehat{\boldsymbol{\beta}}^{k}$ for the above four convex penalties in Step 3 of Algorithm 1, and postpone their derivations to the Appendix. Here $S(a,b) = \operatorname{sign}(a)\cdot(|a| - b)_+$ denotes the shrinkage operator, $\operatorname{sign}(\cdot)$ is the sign function and $(c)_+ = \max(c, 0)$. For all the four penalty functions, the dominant computational effort of each LAMM update is a relatively cheap matrix-vector multiplication involving $\mathbf{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)^T$, thus, with a complexity $O(np)$.

4. Numerical Studies

In this section, we perform extensive numerical studies to evaluate the performance of the LAMM algorithm (Algorithm 1) for fitting penalized SQR with four convex penalties, the lasso (ℓ_1 penalty), elastic net, group lasso, and sparse group lasso. We implement Algorithm 1 using the Gaussian kernel. The numerical performance under other commonly used kernels, such the logistic kernel, Laplacian kernel and uniform kernel, are quite similar and thus we omit the correspondent results. As suggested in Tan, Wang, and Zhou (2022), we set the default bandwidth value as $h = \max\{0.05, \sqrt{\tau(1-\tau)}(\log p/n)^{1/4}\}$ throughout the numerical studies. The empirical evidence from He et al. (2023) and Tan, Wang, and Zhou (2022) shows that the SQR estimator is not susceptible to the choice of h in a reasonable range that is neither too small nor too large. In Section 4.1, we fit penalized SQR with the ℓ_1 and elastic net penalties on simulated data with sparse regression coefficients. We also evaluate the computational efficiency of Algorithm 1, implemented via the `conquer` package, by comparing it to several state-of-the-art packages on penalized regression. In Section 4.2, we fit penalized

Algorithm 1 The LAMM Algorithm for Solving (3.1).

Input: kernel function $K(\cdot)$, penalty function $P(\cdot)$, regularization parameters, bandwidth h, inflation factor $\gamma = 1.2$, and convergence criterion ϵ.

Initialization: $\widehat{\boldsymbol{\beta}}^0 \leftarrow \mathbf{0}$, $\phi_0 \leftarrow 0.01$.

Repeat the following steps until the stopping criterion $\|\widehat{\boldsymbol{\beta}}^k - \widehat{\boldsymbol{\beta}}^{k-1}\|_2 \le \epsilon$ is met, where $\widehat{\boldsymbol{\beta}}^k$ is the kth iterate.

1. Set $\phi_k \leftarrow \max\{\phi_0, \phi_{k-1}/\gamma\}$.
2. **repeat**
3. for $j = 1, \ldots, p$ (or $g = 1, \ldots, G$ for group lasso and sparse group lasso), update $\widehat{\beta}_j^k$ (or $\widehat{\boldsymbol{\beta}}_g^k$) as follows:

weighted lasso	$\widehat{\beta}_j^k \leftarrow S(\widehat{\beta}_j^{k-1} - \phi_k^{-1}\nabla_{\beta_j} Q(\widehat{\boldsymbol{\beta}}^{k-1}), \phi_k^{-1}\lambda_j)$.
elastic net	$\widehat{\beta}_j^k \leftarrow \frac{1}{1+2\phi_k^{-1}\lambda(1-\alpha)} S(\widehat{\beta}_j^{k-1} - \phi_k^{-1}\nabla_{\beta_j} Q(\widehat{\boldsymbol{\beta}}^{k-1}), \phi_k^{-1}\lambda\alpha)$.
group lasso	$\widehat{\boldsymbol{\beta}}_g^k \leftarrow (\widehat{\boldsymbol{\beta}}_g^{k-1} - \phi_k^{-1}\nabla_{\boldsymbol{\beta}_g} Q(\widehat{\boldsymbol{\beta}}^{k-1}))(1 - \frac{\lambda w_g}{\phi_k \|\widehat{\boldsymbol{\beta}}_g^{k-1} - \phi_k^{-1}\nabla_{\boldsymbol{\beta}_g} Q(\widehat{\boldsymbol{\beta}}^{k-1})\|_2})_+$.
sparse group lasso	$\widehat{\boldsymbol{\beta}}_g^k \leftarrow S(\widehat{\boldsymbol{\beta}}_g^{k-1} - \phi_k^{-1}\nabla_{\boldsymbol{\beta}_g} Q(\widehat{\boldsymbol{\beta}}^{k-1}), \phi_k^{-1}\lambda)(1 - \frac{\lambda w_g}{\phi_k \|\widehat{\boldsymbol{\beta}}_g^{k-1} - \phi_k^{-1}\nabla_{\boldsymbol{\beta}_g} Q(\widehat{\boldsymbol{\beta}}^{k-1})\|_2})_+$.

4. **If** $F(\widehat{\boldsymbol{\beta}}^k | \phi_k, \widehat{\boldsymbol{\beta}}^{k-1}) < Q(\widehat{\boldsymbol{\beta}}^k)$, update $\phi_k \leftarrow \gamma\phi_k$.
5. **until** $F(\widehat{\boldsymbol{\beta}}^k | \phi_k, \widehat{\boldsymbol{\beta}}^{k-1}) \ge Q(\widehat{\boldsymbol{\beta}}^k)$.

Output: the updated parameter $\widehat{\boldsymbol{\beta}}^k$.

SQR with the group lasso penalty on simulated data for which the groups of regression coefficients are sparse.

4.1. Simulated Data with Sparse Regression Coefficients

We start with generating $\widetilde{\boldsymbol{x}}_i \in \mathbb{R}^p$ from a multivariate normal distribution $N_p(\mathbf{0}, \boldsymbol{\Sigma})$, where $\boldsymbol{\Sigma} = (0.7^{|j-k|})_{1\le j,k\le p}$, and set $\boldsymbol{x}_i = (1, \widetilde{\boldsymbol{x}}_i^T)^T$. Given $\tau \in (0,1)$, we generate the response y_i from the following linear heteroscedastic model:

$$y_i = \boldsymbol{x}_i^T \boldsymbol{\beta}^* + (0.5|x_{i,p+1}| + 1)\{\epsilon_i - F_{\epsilon_i}^{-1}(\tau)\}, \qquad (4.1)$$

where $F_{\epsilon_i}^{-1}(\tau)$ denotes the τth quantile of the noise variable ϵ_i. We consider two noise distributions: (i) Normal distribution with mean zero and variance 2, that is, $\epsilon_i \sim N(0,2)$, and (ii) t-distribution with 1.5 degrees of freedom, that is, $\epsilon_i \sim t_{1.5}$. Moreover, the vector of regression coefficients $\boldsymbol{\beta}^*$ takes the following two forms: (i) sparse $\boldsymbol{\beta}^*$ with $\beta_1^* = 4$ (intercept), $\beta_2^* = 1.8$, $\beta_4^* = 1.6$, $\beta_6^* = 1.4$, $\beta_8^* = 1.2$, $\beta_{10}^* = 1$, $\beta_{12}^* = -1$, $\beta_{14}^* = -1.2$, $\beta_{16}^* = -1.4$, $\beta_{18}^* = -1.6$, $\beta_{20}^* = -1.8$, and $\beta_j^* = 0$ for all other j's, and (ii) dense $\boldsymbol{\beta}^*$ with $\beta_1^* = 4$ (intercept), $\beta_j^* = 0.8$ for $j = 2, \ldots, 100$, and 0 otherwise.

We compare the proposed algorithm for penalized SQR with the ℓ_1 and elastic net penalties to the ℓ_1-penalized QR implemented by the R package `rqPen`[1] (Sherwood and Maidman 2020). Note that the ADMM-based algorithm proposed by Gu et al. (2018), implemented in the R package `FHDQR`, is incompatible with the current version of R and hence is not included in this article. The regularization parameter λ is selected via 10-fold cross-validation for which the validation error is defined through the quantile loss. We set the additional tuning parameter α for the elastic net penalty as $\alpha = \{0.3, 0.5, 0.7\}$. To evaluate the statistical performance of different methods, we report the estimation error under the ℓ_2-norm, that is, $\|\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*\|_2$, as well as the true positive rate (TPR) and false positive rate (FPR), which are defined as the proportion of correctly estimated non-zeros and the proportion of falsely estimated non-zeros, respectively. The results for the sparse and dense $\boldsymbol{\beta}^*$, averaged over 100 replications, are reported in Tables 1 and 2, respectively.

From Table 1 we see that when true signals are genuinely sparse, both ℓ_1-penalized SQR and QR outperform the elastic net SQR (with different α values) in all three facets. Due to the large number of zeros in $\boldsymbol{\beta}^*$, the performance of the elastic net SQR deteriorates as α decreases, where $\alpha \in [0,1]$ is a user-specified parameter that balances the ℓ_1 penalty and the ridge (ℓ_2) penalty. Table 2 summarizes the results under a dense $\boldsymbol{\beta}^*$ that contains 100 non-zeros coordinates. In this case, the elastic net SQR estimators tend to have lower estimation error and high true positive rate, suggesting that the elastic net penalty may be beneficial when the signals are dense and the signal-to-noise ratio is relatively low.

Furthermore, we provide speed comparisons of the three `R` packages, `glmnet`, `rqPen`, and `conquer`, for fitting sparse linear models. As a benchmark, `glmnet` is used to compute the ℓ_1-penalized least squares (lasso) estimator, whereas `rqPen` and `conquer` are used to fit penalized quantile regression with $\tau = 0.5$. For each method, the regularization parameter is selected from a sequence of 50 λ-values via 10-fold cross-validation. The curves in panels (a) and (c) of Figure 1 represent the estimation error (under ℓ_2 norm) as a function of dimension p, and the curves in panels (b) and (d) of Figure 1 represent the logarithmic computational time (in log second) as a function of dimension p. The sample size n is taken to be $p/2$. As the sample size n increases, the lasso, QR-lasso, and SQR-lasso estimators demonstrate similar convergence rates under normal errors. It is worth noting that the lasso estimator becomes inconsistent in high dimensions when the error distribution follows a $t_{1.5}$-distribution; see Section 5.1 of Loh (2017). This phenomenon explains the non-decaying error curve observed for `glmnet` in plot (c). Notably, the SQR-lasso may even exhibit slight improvements over the QR-lasso, suggesting that incorporating smoothing techniques has the potential to enhance finite-sample performance. Regarding computational efficiency, our `conquer` package showcases significant advancements compared to `rqPen`, particularly when dealing with a large number of predictors. The algorithms implemented in `rqPen` are either a

[1] The `rqPen` package does not have the elastic net penalty option.

Table 1. Numerical comparisons under the linear heteroscedastic model (4.1) with sparse regression coefficients in moderate-dimensional ($n = 500, p = 250$) and high-dimensional settings ($n = 250, p = 500$).

Linear heteroscedastic model with sparse $\boldsymbol{\beta}^*$ and $\tau = 0.5$							
		$(n = 500, p = 250)$			$(n = 250, p = 500)$		
Noise	Methods	Error	TPR	FPR	Error	TPR	FPR
$N(0, 2)$	`SQR` (lasso)	0.852 (0.015)	1 (0)	0.126 (0.007)	0.992 (0.019)	1 (0)	0.071 (0.004)
	`SQR` (elastic net, $\alpha = 0.7$)	1.386 (0.017)	1 (0)	0.268 (0.009)	1.666 (0.018)	1 (0)	0.162 (0.005)
	`SQR` (elastic net, $\alpha = 0.5$)	1.727 (0.015)	1 (0)	0.393 (0.009)	2.108 (0.016)	1 (0)	0.258 (0.007)
	`SQR` (elastic net, $\alpha = 0.3$)	2.056 (0.014)	1 (0)	0.589 (0.010)	2.506 (0.012)	1 (0)	0.416 (0.007)
	`rqPen` (lasso)	0.892 (0.017)	1 (0)	0.135 (0.007)	1.030 (0.020)	1 (0)	0.072 (0.003)
$t_{1.5}$	`SQR` (lasso)	0.758 (0.016)	1 (0)	0.099 (0.005)	0.903 (0.022)	0.980 (0.014)	0.065 (0.004)
	`SQR` (elastic net, $\alpha = 0.7$)	1.293 (0.018)	1 (0)	0.264 (0.007)	1.635 (0.030)	0.980 (0.014)	0.161 (0.005)
	`SQR` (elastic net, $\alpha = 0.5$)	1.676 (0.018)	1 (0)	0.389 (0.008)	2.104 (0.034)	0.980 (0.014)	0.242 (0.006)
	`SQR` (elastic net, $\alpha = 0.3$)	2.058 (0.016)	1 (0)	0.567 (0.008)	2.530 (0.038)	0.980 (0.014)	0.402 (0.009)
	`rqPen` (lasso)	0.793 (0.017)	1 (0)	0.109 (0.006)	0.935 (0.023)	0.980 (0.014)	0.069 (0.003)
Linear heteroscedastic model with sparse $\boldsymbol{\beta}^*$ and $\tau = 0.7$							
$N(0, 2)$	`SQR` (lasso)	0.873 (0.017)	1 (0)	0.112 (0.005)	1.049 (0.020)	0.999 (0.001)	0.064 (0.003)
	`SQR` (elastic net, $\alpha = 0.7$)	1.423 (0.018)	1 (0)	0.247 (0.007)	1.743 (0.021)	1 (0)	0.155 (0.006)
	`SQR` (elastic net, $\alpha = 0.5$)	1.772 (0.016)	1 (0)	0.382 (0.009)	2.169 (0.017)	1 (0)	0.244 (0.006)
	`SQR` (elastic net, $\alpha = 0.3$)	2.113 (0.015)	1 (0)	0.574 (0.010)	2.557 (0.013)	1 (0)	0.401 (0.007)
	`rqPen` (lasso)	0.902 (0.018)	1 (0)	0.120 (0.005)	1.098 (0.020)	0.999 (0.001)	0.065 (0.003)
$t_{1.5}$	`SQR` (lasso)	0.919 (0.020)	1 (0)	0.106 (0.005)	1.104 (0.028)	0.979 (0.014)	0.061 (0.003)
	`SQR` (elastic net, $\alpha = 0.7$)	1.499 (0.020)	1 (0)	0.247 (0.007)	1.849 (0.034)	0.979 (0.014)	0.147 (0.005)
	`SQR` (elastic net, $\alpha = 0.5$)	1.887 (0.020)	1 (0)	0.368 (0.008)	2.313 (0.038)	0.979 (0.014)	0.220 (0.006)
	`SQR` (elastic net, $\alpha = 0.3$)	2.263 (0.016)	1 (0)	0.524 (0.008)	2.706 (0.041)	0.979 (0.014)	0.370 (0.008)
	`rqPen` (lasso)	0.951 (0.021)	1 (0)	0.112 (0.006)	1.124 (0.028)	0.979 (0.014)	0.064 (0.003)

NOTE: The mean (and standard error) of the estimation error under ℓ_2-norm, true and false positive rates (TPR and FPR), averaged over 100 replications, are reported.

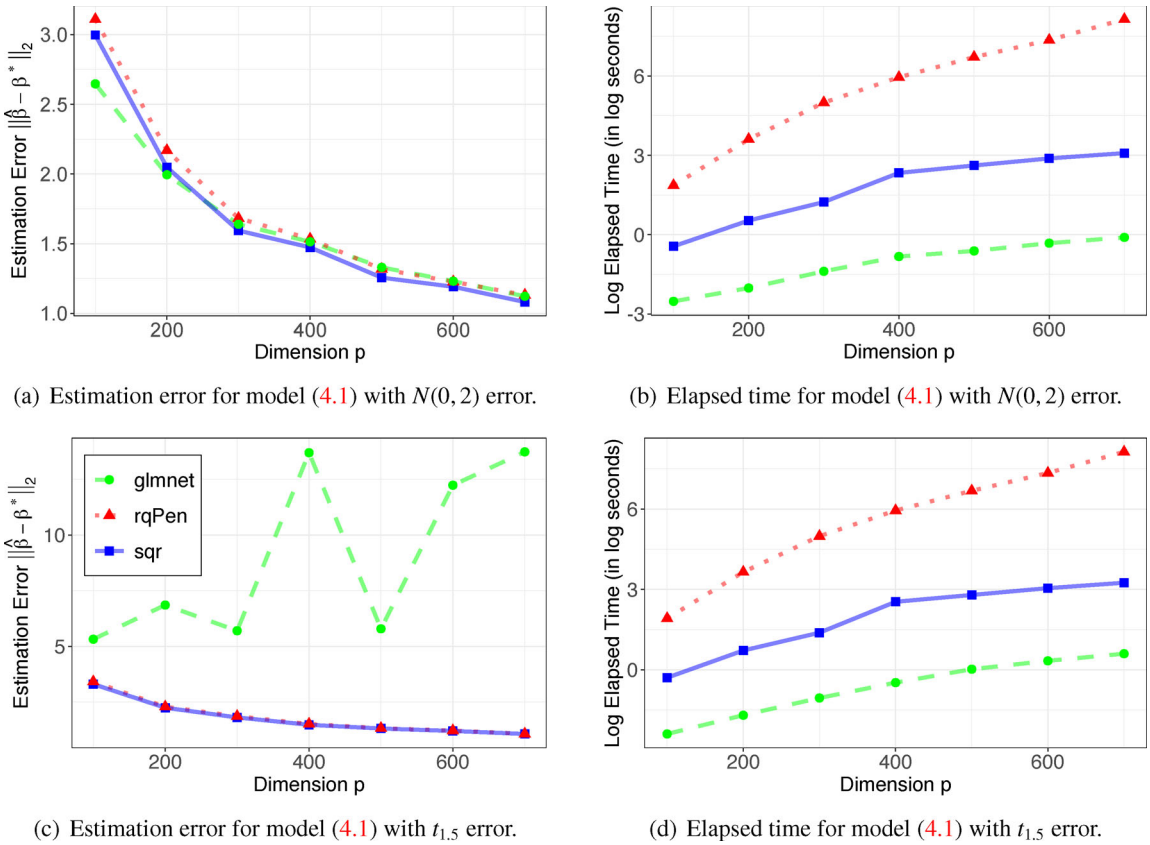
Table 2. Numerical comparisons under the linear heteroscedastic model (4.1) with dense regression coefficients.

Linear heteroscedastic model with dense $\boldsymbol{\beta}^*$ and $\tau = 0.5$							
		$(n = 500, p = 250)$			$(n = 250, p = 500)$		
Noise	Methods	Error	TPR	FPR	Error	TPR	FPR
$N(0, 2)$	`SQR` (lasso)	2.126 (0.019)	1 (0)	0.244 (0.009)	2.600 (0.025)	0.999 (0)	0.163 (0.007)
	`SQR` (elastic net, $\alpha = 0.7$)	1.691 (0.015)	1 (0)	0.312 (0.008)	1.955 (0.019)	1 (0)	0.230 (0.006)
	`SQR` (elastic net, $\alpha = 0.5$)	1.598 (0.016)	1 (0)	0.452 (0.009)	1.898 (0.020)	1 (0)	0.361 (0.006)
	`SQR` (elastic net, $\alpha = 0.3$)	1.633 (0.017)	1 (0)	0.647 (0.006)	2.102 (0.021)	1 (0)	0.584 (0.005)
	`rqPen` (lasso)	2.221 (0.022)	1 (0)	0.245 (0.011)	2.731 (0.025)	0.998 (0)	0.160 (0.006)
$t_{1.5}$	`SQR` (lasso)	2.320 (0.025)	1 (0)	0.231 (0.007)	3.095 (0.083)	0.975 (0.014)	0.156 (0.007)
	`SQR` (elastic net, $\alpha = 0.7$)	1.767 (0.017)	1 (0)	0.314 (0.007)	2.058 (0.038)	0.980 (0.014)	0.222 (0.006)
	`SQR` (elastic net, $\alpha = 0.5$)	1.660 (0.016)	1 (0)	0.447 (0.006)	2.018 (0.037)	0.980 (0.014)	0.356 (0.007)
	`SQR` (elastic net, $\alpha = 0.3$)	1.709 (0.018)	1 (0)	0.641 (0.006)	2.249 (0.040)	0.980 (0.014)	0.567 (0.010)
	`rqPen` (lasso)	2.377 (0.026)	1 (0)	0.248 (0.009)	3.051 (0.059)	0.974 (0.014)	0.146 (0.005)
Linear heteroscedastic model with dense $\boldsymbol{\beta}^*$ and $\tau = 0.7$							
$N(0, 2)$	`SQR` (lasso)	2.214 (0.020)	1 (0)	0.244 (0.009)	2.731 (0.026)	0.998 (0.001)	0.150 (0.006)
	`SQR` (elastic net, $\alpha = 0.7$)	1.725 (0.015)	1 (0)	0.300 (0.008)	2.030 (0.019)	1 (0)	0.228 (0.006)
	`SQR` (elastic net, $\alpha = 0.5$)	1.636 (0.020)	1 (0)	0.442 (0.009)	1.932 (0.018)	1 (0)	0.348 (0.006)
	`SQR` (elastic net, $\alpha = 0.3$)	1.670 (0.018)	1 (0)	0.641 (0.006)	2.100 (0.018)	1 (0)	0.563 (0.005)
	`rqPen` (lasso)	2.326 (0.020)	1 (0)	0.238 (0.010)	2.916 (0.027)	0.996 (0.001)	0.147 (0.006)
$t_{1.5}$	`SQR` (lasso)	2.781 (0.039)	0.998 (0.001)	0.236 (0.008)	3.582 (0.077)	0.968 (0.014)	0.157 (0.006)
	`SQR` (elastic net, $\alpha = 0.7$)	2.005 (0.025)	1 (0)	0.310 (0.008)	2.313 (0.044)	0.980 (0.014)	0.225 (0.007)
	`SQR` (elastic net, $\alpha = 0.5$)	1.849 (0.022)	1 (0)	0.439 (0.007)	2.200 (0.040)	0.980 (0.014)	0.351 (0.007)
	`SQR` (elastic net, $\alpha = 0.3$)	1.892 (0.021)	1 (0)	0.646 (0.007)	2.415 (0.044)	0.980 (0.014)	0.568 (0.010)
	`rqPen` (lasso)	2.836 (0.038)	0.996 (0.001)	0.245 (0.009)	3.583 (0.067)	0.962 (0.014)	0.138 (0.005)

NOTE: Other details are as in Table 1

variant of the Barrodale and Roberts simplex method (Koenker 2023) or an iterative coordinate descent method (Peng and Wang 2015). The plots (b) and (d) demonstrate a substantial reduction in the computational efficiency gap between implementing penalized least squares and penalized quantile regression. We also compared our proposed algorithm to ADMM for fitting ℓ_1-penalized quantile regression, implemented using the R package `FHDQR`, and found that our proposed algorithm is more computationally efficient than that of Gu et al. (2018) when p is large. See Figure B1 in Appendix B for more details.

We proceed with a sensitivity analysis of our proposed algorithm for fitting penalized SQR by varying the smoothing bandwidth h. In this analysis, we maintain $(n, p) = (150, 300)$ and consider the setting outlined in (4.1), involving either sparse regression coefficients or sparse groups of regression coefficients. As before, the random noise follows either the normal distribution $N(0, 2)$ or the $t_{1.5}$-distribution. We conduct



(a) Estimation error for model (4.1) with $N(0, 2)$ error.

(b) Elapsed time for model (4.1) with $N(0, 2)$ error.

(c) Estimation error for model (4.1) with $t_{1.5}$ error.

(d) Elapsed time for model (4.1) with $t_{1.5}$ error.

Figure 1. Estimation error and logarithmic elapsed time (in seconds) under model (4.1) with $N(0, 2)$ and $t_{1.5}$ random noise and $\tau = 0.5$, averaged over 50 datasets for three different algorithms: (i) the proposed algorithm for fitting ℓ_1-penalized SQR with Gaussian kernel; (ii) the ℓ_1-penalized QR implemented using `rqPen`; and (iii) the ℓ_1-penalized least squares method implemented using `glmnet`.

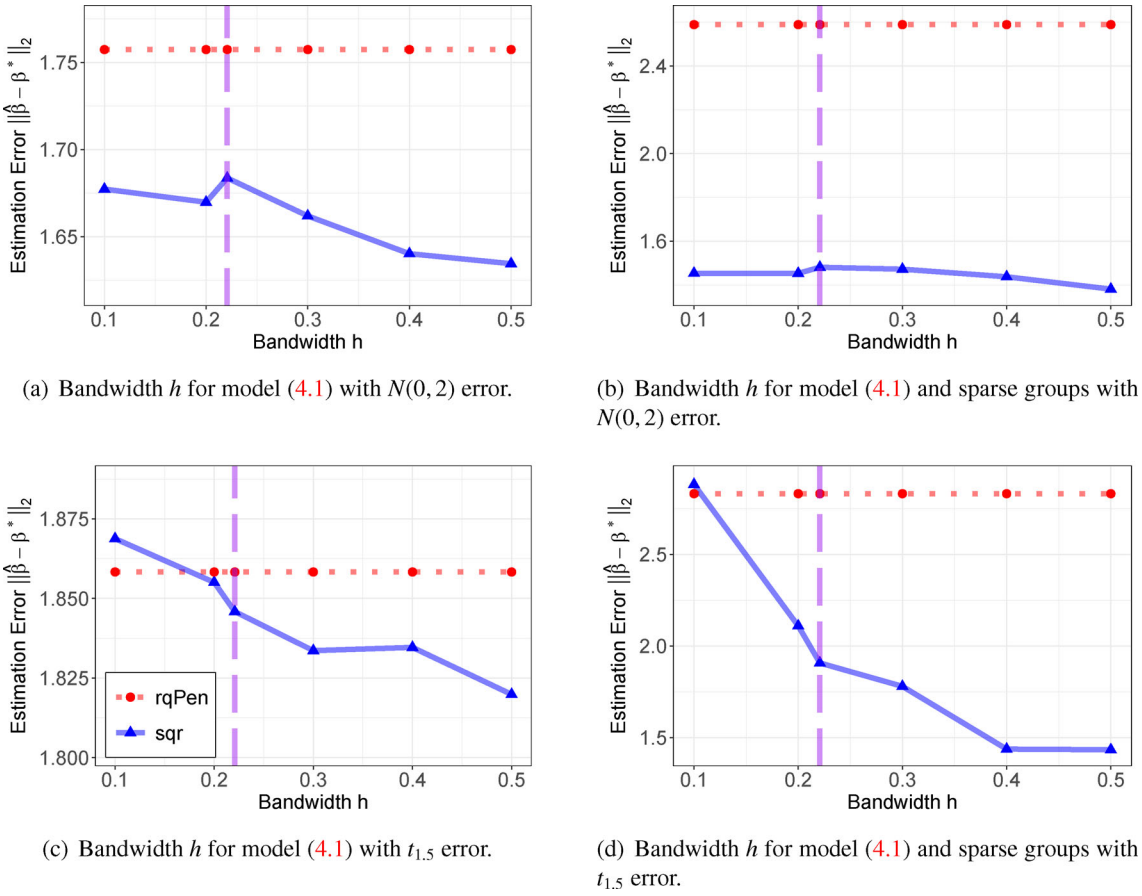
ℓ_1-penalized SQR for sparse regression coefficients and group lasso penalized SQR for sparse groups, selecting h from a range while keeping $\gamma = 1.2$ fixed. We then compare the estimation error with that of ℓ_1-penalized QR, as illustrated in Figure 2. The results indicate that the estimation error of penalized SQR is consistently lower than or comparable to that of ℓ_1-penalized QR across a range of h values, including our default choice. This finding suggests that penalized SQR exhibits robustness to variations in the bandwidth parameter h.

4.2. Simulated Data with Sparse Groups of Regression Coefficients

To further evaluate the performance of penalized SQR with the group lasso penalty, we conduct a comparison between the `conquer` package and the `SGL` package in `R`. The latter is designed to fit regularized linear models (using squared loss), logistic models and Cox models, using a combination of lasso and group lasso penalties. To this date, we are not aware of any existing R package that implements group lasso penalized quantile regression. The regularization parameter λ is once again selected by 10-fold cross-validation, and the weights $w_1, \ldots, w_G$ are set as $w_g = \sqrt{p_g}$, where p_g is the dimension of the sub-vector $\boldsymbol{\beta}_g$.

We generate the data according to (4.1) with 10 groups of regression coefficients $\boldsymbol{\beta}^*$. Specifically, we construct a block-diagonal covariance matrix $\boldsymbol{\Sigma}' = \text{diag}(\boldsymbol{\Sigma}_1, \ldots, \boldsymbol{\Sigma}_{10})$, where $\boldsymbol{\Sigma}_1, \boldsymbol{\Sigma}_2 \in \mathbb{R}^{5\times 5}$, $\boldsymbol{\Sigma}_3, \boldsymbol{\Sigma}_4, \boldsymbol{\Sigma}_5 \in \mathbb{R}^{10\times 10}$, and $\boldsymbol{\Sigma}_6, \ldots, \boldsymbol{\Sigma}_{15} \in \mathbb{R}^{(p-40)/10\times(p-40)/10}$, and each block is an exchangeable covariance matrix with diagonal 1 and off-diagonal elements 0.6. We then generate the covariates from the multivariate normal distribution, $\widetilde{\boldsymbol{x}}_i \sim N_p(\mathbf{0}, \boldsymbol{\Sigma}')$ and set $\boldsymbol{x}_i = (1, \widetilde{\boldsymbol{x}}_i^T)^T$. We construct $\boldsymbol{\beta}^*$ that has a sparse group structure, that is, $\boldsymbol{\beta}^* = \{\beta_0^*, (\boldsymbol{\beta}_1^*)^T, \ldots, (\boldsymbol{\beta}_{15}^*)^T\}^T$ with $\beta_0^* = 4$ (intercept), $\boldsymbol{\beta}_1^* = \mathbf{2} \in \mathbb{R}^5$, $\boldsymbol{\beta}_2^* = \mathbf{1.6} \in \mathbb{R}^5$, $\boldsymbol{\beta}_3^* = -\mathbf{2} \in \mathbb{R}^{10}$, $\boldsymbol{\beta}_4^* = \mathbf{1} \in \mathbb{R}^{10}$, $\boldsymbol{\beta}_5^* = \mathbf{0.6} \in \mathbb{R}^{10}$, and $\boldsymbol{\beta}_6^* = \cdots = \boldsymbol{\beta}_{15}^* = \mathbf{0} \in \mathbb{R}^{10}$.

To assess the performance of group lasso SQR, we calculate the group TPR and group FPR, defined as the proportion of groups that are correctly estimated to contain non-zeros, and the proportion of groups that are incorrectly estimated to contain non-zeros, respectively. Since ℓ_1-penalized methods do not induce group structures, the group TPR and FPR are not well defined. Simulation results under $N(0, 2)$ and $t_{1.5}$ distributed error models, averaged over 100 replications, are reported in Table 3. Under sparse group structures, the group lasso SQR demonstrates comparable performance to its least squares counterpart (implemented via `SGL`) in the presence of Gaussian errors. Additionally, it showcases robustness when confronted with heavy-tailed distributions, as evidenced by the results obtained with $t_{1.5}$-errors. In the latter case, the consistency of penalized least squares estimators is compromised. Our findings suggest that employing a group lasso penalty can be advantageous when the covariates are highly correlated within predefined groups. Moreover, the `conquer` package serves as a useful complement to `SGL`, offering an efficient implementation of group lasso regularized quantile regression.



(a) Bandwidth h for model (4.1) with $N(0, 2)$ error.

(b) Bandwidth h for model (4.1) and sparse groups with $N(0, 2)$ error.

(c) Bandwidth h for model (4.1) with $t_{1.5}$ error.

(d) Bandwidth h for model (4.1) and sparse groups with $t_{1.5}$ error.

Figure 2. Sensitivity analysis of ℓ_1-penalized conquer with a range of bandwidth parameter h. Results for $(n, p) = (150, 300)$ under model (4.1) with sparse regression coefficients and sparse groups of regression coefficients $N(0, 2)$ and $t_{1.5}$ random noise and $\tau = 0.5$, averaged over 100 datasets, with the proposed algorithm for fitting ℓ_1-penalized and group lasso penalized SQR implemented using a Gaussian kernel. The purple vertical dashed line represents our default choice of $h = \max\{0.05, \sqrt{\tau(1-\tau)}(\log p/n)^{1/4}\}$, and the red horizontal dashed line represents the estimation error of ℓ_1-penalized QR.

5. Fused Lasso Additive SQR and World Happiness Data

In this section, we employ the fused lasso additive smoothed quantile regression for flexible and interpretable modeling of the conditional relationship between country happiness level and a set of covariates, using the world happiness data (United Nations Development Programme 2012; World Bank Group 2012; Helliwell, Layard, and Sachs 2013) previously studied in Petersen, Witten, and Simon (2016). Specifically, the goal is to study the conditional distribution of country-level happiness index, the average of Cantril Scale (Cantril 1965) responses of approximately 3000 residents in each country, given 12 country-level predictors, including but not limited to log gross national income (USD), satisfaction with freedom of choice, satisfaction with job, satisfaction with community, and trust in national government.

We first provide a brief overview of the fused lasso additive model in Petersen, Witten, and Simon (2016). The fused lasso additive model seeks to balance interpretability and flexibility by approximating the additive function for each covariate via a piecewise constant function. Let $\boldsymbol{\theta}_j = (\theta_{1j}, \ldots, \theta_{nj})^T$ for $j = 1, \ldots, p$ and let $\mathbf{D}$ be an $(n-1) \times n$ matrix with entries $D_{ii} = 1$ and $D_{i(i+1)} = -1$ for $i = 1, \ldots, n-1$, and $D_{ij} = 0$ for $j \neq i, i+1$. Moreover, let $\mathbf{P}_j$ be a permutation matrix that orders the elements of $\boldsymbol{x}_j = (x_{1j}, \ldots, x_{nj})^T$ from least to greatest. The fused lasso additive model estimator in Petersen, Witten, and Simon (2016) can be obtained by solving the convex optimization problem

$$\underset{\theta_0 \in \mathbb{R}, \boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_p \in \mathbb{R}^n}{\text{minimize}} \ \frac{1}{n} \sum_{i=1}^{n} \left(y_i - \theta_0 - \sum_{j=1}^{p} \theta_{ij} \right)^2 + \lambda \sum_{j=1}^{p} \|\mathbf{D}\mathbf{P}_j \boldsymbol{\theta}_j\|_1,$$
$$\text{subject to } \mathbb{1}^T \boldsymbol{\theta}_j = 0, \ j = 1, \ldots, p, \tag{5.1}$$

where $\|\mathbf{D}\mathbf{P}_j \boldsymbol{\theta}_j\|_1$ is a fused lasso type penalty that encourages the consecutive entries of the ordered parameters $\mathbf{P}_j \boldsymbol{\theta}_j$ to be the same. We refer the reader to Petersen and Witten (2019), Wu and Witten (2019), and Sadhanala and Tibshirani (2019) for a summary of recent work on flexible and interpretable additive models.

We now propose the fused lasso additive smoothed quantile regression for interpretable and flexible modeling of the conditional distribution of y given $\boldsymbol{x}$ at specific quantile levels. Specifically, we propose to solve the following optimization problem by substituting the squared error loss in (5.1) via the smoothed quantile loss in (2.4):

$$\underset{\theta_0 \in \mathbb{R}, \boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_p \in \mathbb{R}^n}{\text{minimize}} \ \frac{1}{n} \sum_{i=1}^{n} \ell_{h,\tau} \left(y_i - \theta_0 - \sum_{j=1}^{p} \theta_{ij} \right) + \lambda \sum_{j=1}^{p} \|\mathbf{D}\mathbf{P}_j \boldsymbol{\theta}_j\|_1$$
$$\text{subject to } \mathbb{1}^T \boldsymbol{\theta}_j = 0, \ j = 1, \ldots, p. \tag{5.2}$$

Table 3. Numerical comparisons under the linear heteroscedastic model (4.1) with groups of regression coefficients in moderate-dimensional ($n = 500, p = 250$) and high-dimensional settings ($n = 250, p = 500$).

Linear heteroscedastic model with sparse groups of $\boldsymbol{\beta}^*$ and $\tau = 0.5$							
		$(n = 500, p = 250)$			$(n = 250, p = 500)$		
Noise	Methods	Error	Group TPR	Group FPR	Error	Group TPR	Group FPR
$N(0, 2)$	`SQR` (lasso)	1.114 (0.013)	1 (0)	0.022 (0.002)	1.286 (0.016)	1 (0)	0.017 (0.002)
	`rqPen` (lasso)	1.161 (0.015)	1 (0)	0.107 (0.008)	1.356 (0.019)	1 (0)	0.074 (0.006)
	`SQR` (group)	0.772 (0.008)	1 (0)	0.020 (0.005)	0.840 (0.009)	1 (0)	0.038 (0.007)
	`SGL` (group)	1.092 (0.042)	1 (0)	0.068 (0.013)	1.280 (0.057)	1 (0)	0.062 (0.010)
$t_{1.5}$	`SQR` (lasso)	1.412 (0.040)	0.996 (0.001)	0.002 (0)	1.562 (0.037)	0.975 (0.014)	0.002 (0)
	`rqPen` (lasso)	1.075 (0.015)	1 (0)	0.102 (0.006)	1.278 (0.026)	0.980 (0.014)	0.059 (0.004)
	`SQR` (group)	0.914 (0.028)	1 (0)	0 (0)	0.953 (0.023)	0.980 (0.014)	0 (0)
	`SGL` (group)	3.376 (0.483)	1 (0)	0.228 (0.034)	2.357 (0.133)	0.980 (0.014)	0.147 (0.025)
Linear heteroscedastic model with sparse groups of $\boldsymbol{\beta}^*$ and $\tau = 0.7$							
$N(0, 2)$	`SQR` (lasso)	1.195 (0.014)	1 (0)	0.025 (0.002)	1.381 (0.019)	1 (0)	0.016 (0.001)
	`rqPen` (lasso)	1.232 (0.014)	1 (0)	0.106 (0.006)	1.432 (0.023)	1 (0)	0.067 (0.004)
	`SQR` (group)	0.809 (0.009)	1 (0)	0.033 (0.010)	0.893 (0.011)	1 (0)	0.048 (0.009)
	`SGL` (group)	1.427 (0.059)	1 (0)	0.061 (0.010)	1.540 (0.077)	1 (0)	0.063 (0.012)
$t_{1.5}$	`SQR` (lasso)	1.764 (0.052)	0.995 (0.001)	0.003 (0)	1.948 (0.051)	0.969 (0.014)	0.003 (0)
	`rqPen` (lasso)	1.253 (0.018)	1 (0)	0.096 (0.006)	1.541 (0.032)	0.978 (0.014)	0.060 (0.004)
	`SQR` (group)	1.155 (0.040)	1 (0)	0.001 (0.001)	1.180 (0.031)	0.980 (0.014)	0.002 (0.001)
	`SGL` (group)	3.529 (0.480)	1 (0)	0.229 (0.034)	2.492 (0.138)	0.980 (0.014)	0.150 (0.025)

NOTE: The mean (and standard error) of the estimation error under ℓ_2-norm, group true and group false positive rates (Group TPR and Group FPR), averaged over 100 replications, are reported.

Optimization problem (5.2) is convex and can be solved by updating $\theta_0 \in \mathbb{R}$ and each block of parameters $\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_p \in \mathbb{R}^n$ iteratively, holding all others fixed. We employ a block coordinate descent algorithm similar to that of Petersen, Witten, and Simon (2016) to solve (5.2), which we outline in Algorithm 2.

Specifically, at the kth iteration, let $r_{ij}^k = y_i - \widehat{\theta}_0^{k-1} - \sum_{j' \neq j} \widehat{\theta}_{ij'}^{k-1}$ be the residuals, where $\widehat{\theta}_0^{k-1}$ and $\widehat{\theta}_{ij}^{k-1}$ are solutions from the $(k-1)$th iteration. For each $j = 1, \ldots, p$, we obtain an update for $\boldsymbol{\theta}_j$, holding the other parameters fixed, by solving the following convex optimization problem

$$\widehat{\boldsymbol{\theta}}_j^k = \text{argmin}_{\boldsymbol{\theta}_j \in \mathbb{R}^n} \frac{1}{n} \sum_{i=1}^n \ell_{h,\tau}(r_{ij}^k - \theta_{ij}) + \lambda \|\mathbf{D}\mathbf{P}_j \boldsymbol{\theta}_j\|_1. \quad (5.3)$$

Tibshirani and Taylor (2011) showed that (5.3) can be rewritten as a lasso problem through some transformation on the regression coefficients, and thus Algorithm 1 can naturally be employed to solve (5.3). To this end, we construct a rank n matrix $\widetilde{\mathbf{D}} = (\mathbf{D}^T, \mathbb{1}^T)^T$ by stacking an n-dimensional vector of ones to $\mathbf{D}$. Let $\boldsymbol{\zeta}_j = \widetilde{\boldsymbol{D}}\mathbf{P}_j\boldsymbol{\theta}_j$. By a change of variable, (5.3) reduces to a lasso-type problem that can be solved using Algorithm 1, that is,

$$\widehat{\boldsymbol{\zeta}}_j^k = \text{argmin}_{\boldsymbol{\zeta}_j \in \mathbb{R}^n} \frac{1}{n} \sum_{i=1}^n \ell_{h,\tau}\left(r_{ij}^k - (\mathbf{P}_j^{-1}\widetilde{\mathbf{D}}^{-1}\boldsymbol{\zeta}_j)_i\right) + \lambda \sum_{i=1}^{n-1} |\zeta_{ij}|, \quad (5.4)$$

where we do not penalize the last coordinate of $\boldsymbol{\zeta}_j$. The updates for $\boldsymbol{\theta}_j$ can then be constructed as $\widehat{\boldsymbol{\theta}}_j^k = \mathbf{P}_j^{-1}\widetilde{\mathbf{D}}^{-1}\widehat{\boldsymbol{\zeta}}_j^k$. We refer the reader to Tibshirani and Taylor (2011) for more details.

We apply the proposed method to investigate the conditional distribution of country-level happiness index at different quantile levels $\tau = \{0.2, 0.5, 0.8\}$. We implement our proposed method under the Gaussian kernel with $h = \max[0.05, \sqrt{\tau(1-\tau)}\{\log(p)/n\}^{1/4}]$, and λ selected using cross-validation. The estimated fits for three selected covariates are shown in Figure 3: the first, second, and third rows in Figure 3 correspond to the results for $\tau = 0.2$, $\tau = 0.5$, and $\tau = 0.8$, respectively. For $\tau = 0.5$ the estimated fits are similar to that of the fused lasso additive model presented in Petersen, Witten, and Simon (2016). In particular, we find that an increased in gross national income, up to a certain level, is associated with increased happiness, conditional on the other predictors. Moreover, the differences in conditional associations between country's happiness level and both gross national income and percent trustful of national government are negligible for the three quantile levels. It is interesting to see that the conditional association between mean happiness index and females with secondary education are different for the three different quantile levels, suggesting that there are potential heterogeneous effects.

Algorithm 2 A Block Coordinate Descent Algorithm for Solving (5.2).

Input: kernel function $K(\cdot)$, regularization parameter λ, smoothing parameter h, the matrix $\widetilde{\mathbf{D}}$, and convergence criterion ϵ.

Initialization: $\widehat{\theta}_0^0 \leftarrow 0, \widehat{\boldsymbol{\theta}}_j^0 \leftarrow \mathbf{0}$ for $j = 1, \ldots, p$.

Iterate: for each $j = 1, \ldots, p$, until the stopping criterion $|\widehat{\theta}_0^k - \widehat{\theta}_0^{k-1}| + \sum_{j=1}^p \|\widehat{\boldsymbol{\theta}}_j^k - \widehat{\boldsymbol{\theta}}_j^{k-1}\|_2 \leq \epsilon$ is met, where $\widehat{\boldsymbol{\theta}}_j^k$ is the value of $\boldsymbol{\theta}_j$ obtained at the kth iteration.

1. Update the residuals $r_{ij}^k \leftarrow y_i - \widehat{\theta}_0^{k-1} - \sum_{j' \neq j} \widehat{\theta}_{ij'}^{k-1}$ for $i = 1, \ldots, n$.
2. Update $\widehat{\boldsymbol{\zeta}}_j^k$ as
$$\widehat{\boldsymbol{\zeta}}_j^k \leftarrow \text{argmin}_{\boldsymbol{\zeta}_j \in \mathbb{R}^n} \frac{1}{n} \sum_{i=1}^n \ell_{h,\tau}\left(r_{ij}^k - (\mathbf{P}_j^{-1}\widetilde{\mathbf{D}}^{-1}\boldsymbol{\zeta}_j)_i\right) + \lambda \sum_{i=1}^{n-1} |\zeta_{ij}|$$
using Algorithm 1.
3. Update the parameters $\widehat{\boldsymbol{\theta}}_j^k \leftarrow \mathbf{P}_j^{-1}\widetilde{\mathbf{D}}^{-1}\widehat{\boldsymbol{\zeta}}_j^k$.
4. Update the intercept $\widehat{\theta}_0^k \leftarrow \widehat{\theta}_0^{k-1} + n^{-1}\sum_{i=1}^n \widehat{\theta}_{ij}^k$.
5. Center the parameters $\widehat{\boldsymbol{\theta}}_j^k \leftarrow \widehat{\boldsymbol{\theta}}_j^k - n^{-1}\sum_{i=1}^n \widehat{\theta}_{ij}^k$.

Output: the updated parameters $\widehat{\theta}_0^k, \widehat{\boldsymbol{\theta}}_1^k, \ldots, \widehat{\boldsymbol{\theta}}_p^k$.

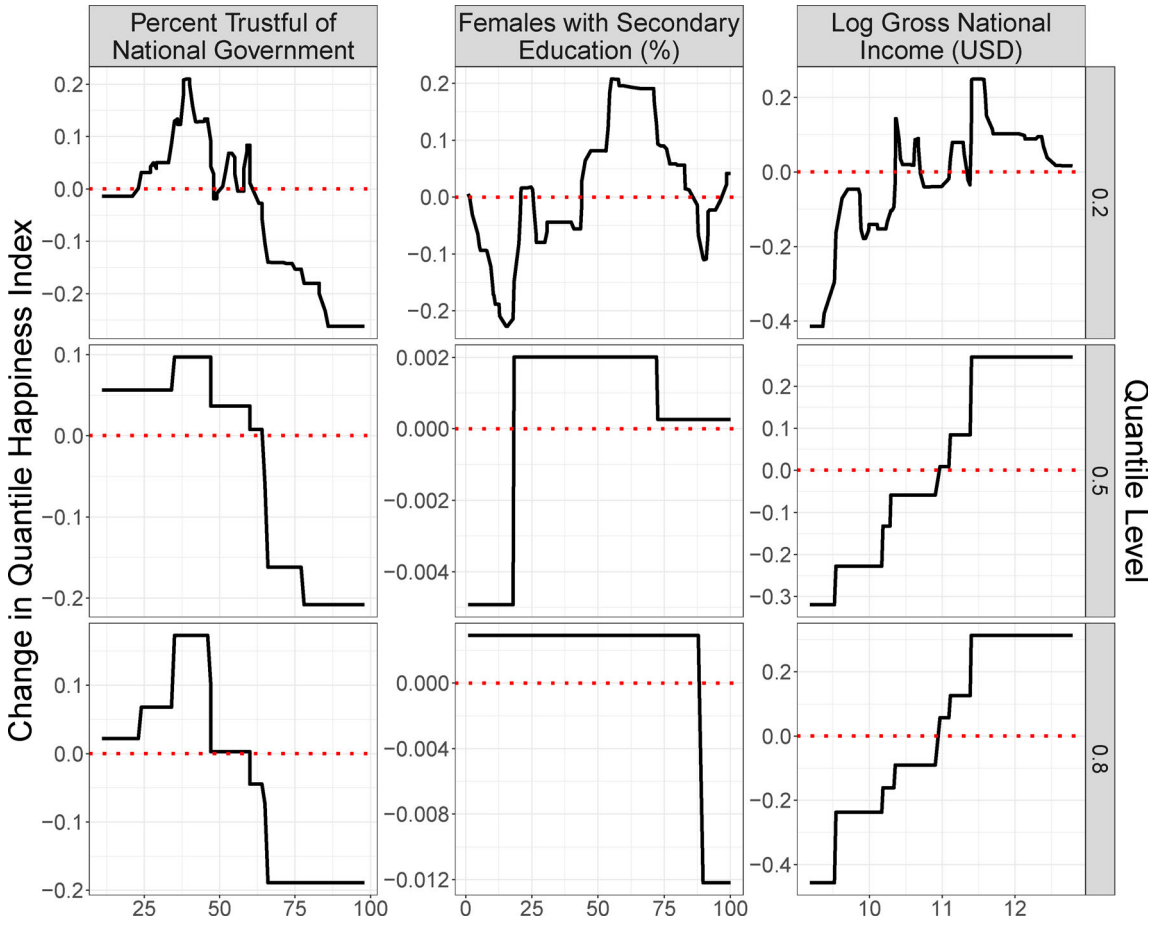


Figure 3. Conditional associations between country's happiness level and three selected country-level predictors for $\tau = \{0.2, 0.5, 0.8\}$. The first, second, and third rows correspond to results for $\tau = 0.2$, $\tau = 0.5$, and $\tau = 0.8$, respectively.

Appendix A. Derivation of Updates for Different Penalty Functions in Algorithm 1

In this section, we provide a detailed derivation of the iterative updates used in Step 3 of Algorithm 1 under the following four convex penalty functions, the weighted lasso (Tibshirani 1996), elastic net (Zou and Hastie 2005), group lasso (Yuan and Lin 2006), and sparse group lasso (Simon et al. 2013) penalty functions.

By the first-order optimization condition and using the isotropic form of $F(\boldsymbol{\beta}|\phi_k, \widehat{\boldsymbol{\beta}}^{k-1})$ in (3.3), the minimizer $\widehat{\boldsymbol{\beta}}^k$ of (3.3) is such that

$$\mathbf{0} \in \nabla_{\boldsymbol{\beta}} Q(\widehat{\boldsymbol{\beta}}^{k-1}) + \phi_k(\widehat{\boldsymbol{\beta}}^k - \widehat{\boldsymbol{\beta}}^{k-1}) + \partial P(\boldsymbol{\beta})|_{\boldsymbol{\beta}=\widehat{\boldsymbol{\beta}}^k}, \tag{A.1}$$

where ∂P is the subdifferential of the convex function $P : \mathbb{R}^p \to [0, \infty)$. Recall that β_1 is the intercept. A common practice is to leave the intercept term unpenalized, that is, $P(\boldsymbol{\beta})$ depends only on $\beta_2, \ldots, \beta_p$ with $\nabla_{\beta_1} P(\boldsymbol{\beta}) = 0$. In this case, the update rule for the intercept takes the form

$$\begin{aligned} &\nabla_{\beta_1} Q(\widehat{\boldsymbol{\beta}}^{k-1}) + \phi_k(\widehat{\beta}_1^k - \widehat{\beta}_1^{k-1}) = 0 \quad \Longleftrightarrow \\ &\widehat{\beta}_1^k = \widehat{\beta}_1^{k-1} - \phi_k^{-1} \nabla_{\beta_1} Q(\widehat{\boldsymbol{\beta}}^{k-1}). \end{aligned}$$

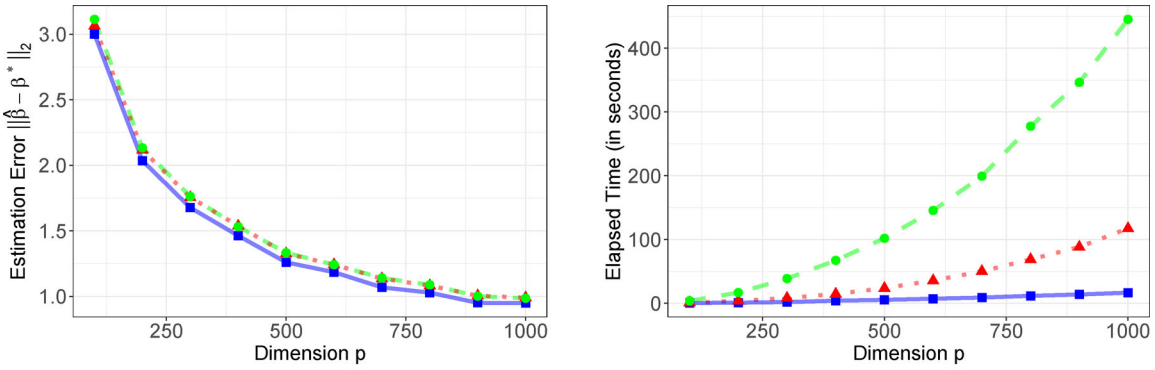
The updates for the regression coefficients $\widehat{\boldsymbol{\beta}}^k$ depend on the specific structure of $\partial P(\boldsymbol{\beta})$, and can be carried out component-wise, or group-wise in the case of group lasso and sparse group lasso penalty functions. Let $S(a, b) = \text{sign}(a) \cdot (|a| - b, 0)_+$ be the soft-thresholding operator, where $\text{sign}(\cdot)$ is the sign function and $(c)_+ = \max(c, 0)$. In addition, let $\nabla_{\boldsymbol{\beta}_g} Q(\widehat{\boldsymbol{\beta}}^{k-1})$ be the sub-vector of the gradient $\nabla_{\boldsymbol{\beta}} Q(\widehat{\boldsymbol{\beta}}^{k-1})$, indexed by the gth group. In the following, we provide explicit derivations for the updates of $\widehat{\boldsymbol{\beta}}^k$ under the four penalty functions.

(1) Weighted lasso (Tibshirani 1996): $P(\boldsymbol{\beta}) = \sum_{j=1}^p \lambda_j |\beta_j|$, where $\lambda_j \geq 0$ for $j = 1, \ldots, p$. The element-wise updates for $\widehat{\beta}_j^k, j = 1, \ldots, p$, are computed as follows:

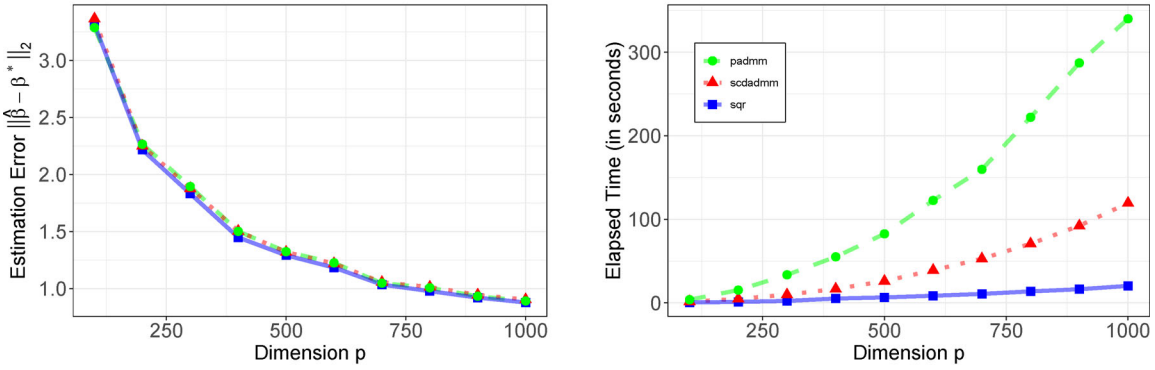
$$\begin{aligned} &\nabla_{\beta_j} Q(\widehat{\boldsymbol{\beta}}^{k-1}) + \phi_k(\widehat{\beta}_j^k - \widehat{\beta}_j^{k-1}) + \lambda_j \,\text{sign}(\widehat{\beta}_j^k) = 0, \\ \Longrightarrow\ &\widehat{\beta}_j^k + \phi_k^{-1}\lambda_j \,\text{sign}(\widehat{\beta}_j^k) = \widehat{\beta}_j^{k-1} - \phi_k^{-1} \nabla_{\beta_j} Q(\widehat{\boldsymbol{\beta}}^{k-1}), \\ \Longrightarrow\ &\widehat{\beta}_j^k = S(\widehat{\beta}_j^{k-1} - \phi_k^{-1} \nabla_{\beta_j} Q(\widehat{\boldsymbol{\beta}}^{k-1}), \phi_k^{-1}\lambda_j). \end{aligned}$$

(2) Elastic net (Zou and Hastie 2005): $P(\boldsymbol{\beta}) = \lambda\alpha\|\boldsymbol{\beta}\|_1 + \lambda(1-\alpha)\|\boldsymbol{\beta}\|_2^2$. The element-wise updates for $\widehat{\beta}_j^k, j = 1, \ldots, p$, are computed as follows:

$$\begin{aligned} &\nabla_{\beta_j} Q(\widehat{\boldsymbol{\beta}}^{k-1}) + \phi_k(\widehat{\beta}_j^k - \widehat{\beta}_j^{k-1}) + \lambda\alpha \,\text{sign}(\widehat{\beta}_j^k) + 2\lambda(1-\alpha)\widehat{\beta}_j^k = 0, \\ \Longrightarrow\ &(2\lambda(1-\alpha) + \phi_k)\widehat{\beta}_j^k + \lambda\alpha \,\text{sign}(\widehat{\beta}_j^k) = \phi_k \widehat{\beta}_j^{k-1} - \nabla_{\beta_j} Q(\widehat{\boldsymbol{\beta}}^{k-1}), \\ \Longrightarrow\ &(1 + 2\phi_k^{-1}\lambda(1-\alpha))\widehat{\beta}_j^k + \phi_k^{-1}\lambda\alpha \,\text{sign}(\widehat{\beta}_j^k) \\ &\quad = \widehat{\beta}_j^{k-1} - \phi_k^{-1} \nabla_{\beta_j} Q(\widehat{\boldsymbol{\beta}}^{k-1}), \\ \Longrightarrow\ &\widehat{\beta}_j^k = \frac{1}{1 + 2\phi_k^{-1}\lambda(1-\alpha)} S(\widehat{\beta}_j^{k-1} - \phi_k^{-1} \nabla_{\beta_j} Q(\widehat{\boldsymbol{\beta}}^{k-1}), \phi_k^{-1}\lambda\alpha). \end{aligned}$$



(a) Estimation error for Model (4.1) with $N(0, 2)$ error. (b) Elapsed time for Model (4.1) with $N(0, 2)$ error.



(c) Estimation error for Model (4.1) with $t_{1.5}$ error. (d) Elapsed time for Model (4.1) with $t_{1.5}$ error.

Figure B1. Estimation error and elapsed time in seconds under Model (4.1) with $N(0, 2)$ error and $t_{1.5}$ random noise and $\tau = 0.5$, averaged over 100 datasets for three different algorithms: (i) the proposed algorithm with the Gaussian kernel for fitting ℓ_1-penalized SQR; (ii) the ℓ_1-penalized QR implemented using the smooth coordinate descent ADMM algorithm in Gu et al. (2018); and (iii) the ℓ_1-penalized QR implemented using the proximal ADMM algorithm in Gu et al. (2018). The sample size n is taken to be $2p$.

(3) Group lasso (Yuan and Lin 2006): $P(\boldsymbol{\beta}) = \lambda \sum_{g=1}^{G} w_g \|\boldsymbol{\beta}_g\|_2$. The updates for $\widehat{\boldsymbol{\beta}}_g^k, g = 1, \ldots, G$, are computed group-wise as follows:

$$\nabla_{\boldsymbol{\beta}_g} Q(\widehat{\boldsymbol{\beta}}^{k-1}) + \phi_k(\widehat{\boldsymbol{\beta}}_g^k - \widehat{\boldsymbol{\beta}}_g^{k-1}) + \lambda w_g \frac{\widehat{\boldsymbol{\beta}}_g^k}{\|\widehat{\boldsymbol{\beta}}_g^k\|_2} = 0,$$

$$\Longrightarrow \widehat{\boldsymbol{\beta}}_g^k + \phi_k^{-1}\lambda w_g \frac{\widehat{\boldsymbol{\beta}}_g^k}{\|\widehat{\boldsymbol{\beta}}_g^k\|_2} = \widehat{\boldsymbol{\beta}}_g^{k-1} - \phi_k^{-1}\nabla_{\boldsymbol{\beta}_g} Q(\widehat{\boldsymbol{\beta}}^{k-1}),$$

$$\Longrightarrow \widehat{\boldsymbol{\beta}}_g^k = (\widehat{\boldsymbol{\beta}}_g^{k-1} - \phi_k^{-1}\nabla_{\boldsymbol{\beta}_g} Q(\widehat{\boldsymbol{\beta}}^{k-1})) \times \left(1 - \frac{\lambda w_g}{\phi_k \|\widehat{\boldsymbol{\beta}}_g^{k-1} - \phi_k^{-1}\nabla_{\boldsymbol{\beta}_g} Q(\widehat{\boldsymbol{\beta}}^{k-1})\|_2}\right)_+.$$

(4) Sparse group lasso (Simon et al. 2013): $P(\boldsymbol{\beta}) = \lambda\|\boldsymbol{\beta}\|_1 + \lambda \sum_{g=1}^{G} w_g \|\boldsymbol{\beta}_g\|_2$. The updates for $\widehat{\boldsymbol{\beta}}_g^k, g = 1, \ldots, G$, are computed group-wise as follows:

$$\nabla_{\boldsymbol{\beta}_g} Q(\widehat{\boldsymbol{\beta}}^{k-1}) + \phi_k(\widehat{\boldsymbol{\beta}}_g^k - \widehat{\boldsymbol{\beta}}_g^{k-1}) + \lambda \operatorname{sign}(\widehat{\boldsymbol{\beta}}_g^k) + \lambda w_g \frac{\widehat{\boldsymbol{\beta}}_g^k}{\|\widehat{\boldsymbol{\beta}}_g^k\|_2} = 0,$$

$$\Longrightarrow \widehat{\boldsymbol{\beta}}_g^k + \phi_k^{-1}\lambda \operatorname{sign}(\widehat{\boldsymbol{\beta}}_g^k) + \phi_k^{-1}\lambda w_g \frac{\widehat{\boldsymbol{\beta}}_g^k}{\|\widehat{\boldsymbol{\beta}}_g^k\|_2} = \widehat{\boldsymbol{\beta}}_g^{k-1} - \phi_k^{-1}\nabla_{\boldsymbol{\beta}_g} Q(\widehat{\boldsymbol{\beta}}^{k-1}),$$

$$\Longrightarrow \widehat{\boldsymbol{\beta}}_g^k = S(\widehat{\boldsymbol{\beta}}_g^{k-1} - \phi_k^{-1}\nabla_{\boldsymbol{\beta}_g} Q(\widehat{\boldsymbol{\beta}}^{k-1}), \phi_k^{-1}\lambda) \times \left(1 - \frac{\lambda w_g}{\phi_k \|\widehat{\boldsymbol{\beta}}_g^{k-1} - \phi_k^{-1}\nabla_{\boldsymbol{\beta}_g} Q(\widehat{\boldsymbol{\beta}}^{k-1})\|_2}\right)_+,$$

where $S(a, b)$ is the soft-thresholding operator, applied element-wise within the gth group.

Appendix B. Additional Numerical Studies

We compared our proposed algorithm for ℓ_1-penalized smoothed quantile regression with the Gaussian kernel and default bandwidth value $h = \max\{0.05, \sqrt{\tau(1-\tau)}(\log p/n)^{1/4}\}$, to ℓ_1-penalized quantile regression implemented using the ADMM algorithm in Gu et al. (2018). We compare our algorithm to both algorithms proposed in Gu et al. (2018), smooth coordinate descent ADMM (scdADMM) and proximal ADMM (pADMM), implemented via the `FHDQR` package. The data are generated according to Model (4.1) with $\tau = 0.5$ and $N(0, 2)$ and $t_{1.5}$ random noise. Results, averaged over 100 datasets, are reported in Figure B1. We see that the proposed algorithm has a significant computational gain especially in high dimensions and achieves comparable estimation error to that of Gu et al. (2018).

Another rapid ADMM-based algorithm, known as QPADM, for penalized quantile regression, was introduced in Yu, Lin, and Wang (2017). However, since there is no readily available package to directly implement this method, we did not include a runtime comparison in this study. The source code for the `conquer` package was crafted using RcppArmadillo. Consequently, comparing it with QPADM implemented in R code would be unfair.

Acknowledgments

We are grateful to the Editor, the Associate Editor, and anonymous reviewers for their constructive comments and suggestions that have significantly improved the article.

Disclosure Statement

No potential conflict of interest was reported by the author(s).

Funding

K. M. Tan was supported by NSF Grants DMS-2113356 and NSF DMS-2238428. W.-X. Zhou acknowledges the support of the NSF Grant DMS-2113409.

ORCID

Wen-Xin Zhou https://orcid.org/0000-0002-2761-485X

References

Beck, A., and Teboulle, M. (2009), "A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems," *SIAM Journal of Imaging Science*, 2, 183–202. [2]

Belloni, A., and Chernozhukov, V. (2011), "ℓ_1-Penalized Quantile Regression in High-Dimensional Sparse Models," *The Annals of Statistics*, 39, 82–130. [3]

Brown, B. M., and Wang, Y. G. (2005), "Standard Errors and Covariance Matrices for Smoothed Rank Estimators," *Biometrika*, 92, 149–158. [2]

Bühlmann, P., and van de Geer, S. (2011), *Statistics for High-Dimensional Data: Methods, Theory and Applications*, Heidelberg: Springer. [1]

Cantril, H. (1965), *The Pattern of Human Concerns*, New Brunswick: Rutgers University Press. [8]

Choi, T., and Choi, S. (2021), "A Fast Algorithm for the Accelerated Failure Time Model with High-Dimensional Time-to-Event Data," *Journal of Statistical Computation and Simulation*, 91, 3385–3403. [2]

Fan, J., and Li, R. (2001), "Variable Selection via Nonconcave Regularized Likelihood and Its Oracle Properties," *Journal of American Statistical Association*, 96, 1348–1360. [1]

Fan, J., Liu, H., Sun, Q., and Zhang, T. (2018), "I-LAMM for Sparse Learning: Simultaneous Control of Algorithmic Complexity and Statistical Error," *The Annals of Statistics*, 46, 814–841. [2,4]

Fernandes, M., Guerre, E., and Horta, E. (2021), "Smoothing Quantile Regressions," *Journal of Business & Economic Statistics*, 39, 338–357. [2,3]

Friedman, J., Hastie, T., and Tibshirani, R. (2010), "Regularization Paths for Generalized Linear Models via Coordinate Descent," *Journal of Statistical Software*, 33, 1–22. [1]

Gu, Y., Fan, J., Kong, L., Ma, S., and Zou, H. (2018), "ADMM for High-Dimensional Sparse Regularized Quantile Regression," *Technometrics*, 60, 319–331. [1,3,5,6,11]

Hastie, T., Tibshirani, R., and Wainwright, M. (2015), *Statistical Learning with Sparsity: The Lasso and Generalizations*, Boca Raton, FL: CRC Press. [1]

He, X., Pan, X., Tan, K. M., and Zhou, W.-X. (2023), "Smoothed Quantile Regression with Large-Scale Inference," *Journal of Econometrics*, 232, 367–388. [2,3,4]

He, X., Pan, X., Tan, K. M., and Zhou, W.-X. (2022), *Package* "`conquer`", version 1.3.0. Reference manual: *https://cran.r-project.org/web/packages/conquer/conquer.pdf*. [2]

Helliwell, J., Layard, R., and Sachs, J. (2013), "World Happiness Report 2013," Sustainable Development Solutions Network. [8]

Hunter, D. R., and Lange, K. (2004), "A Tutorial on MM Algorithms," *The American Statistician*, 58, 30–37. [4]

Horowitz, J. L. (1998), "Bootstrap Methods for Median Regression Models," *Econometrica*, 66, 1327–1351. [2]

Kato, K. (2011), "Group Lasso for High Dimensional Sparse Quantile Regression Models," arXiv preprint arXiv:1103.1458. [2]

Koenker, R. (2005), *Quantile Regression*, Cambridge: Cambridge University Press. [1]

——— (2023), *Package* "`quantreg`", version 5.95. Reference manual: *https://cran.r-project.org/web/packages/quantreg/quantreg.pdf*. [3,6]

Koenker, R., and Bassett, G. (1978), "Regression Quantiles," *Econometrica*, 46, 33–50. [1,2]

Koenker, R., Chernozhukov, V., He, X., and Peng, L. (2017), *Handbook of Quantile Regression*, Boca Raton, FL: CRC Press. [1]

Koenker, R., and Ng, P. (2005), "A Frisch-Newton Algorithm for Sparse Quantile Regression," *Acta Mathematicae Applicatae Sinica*, 21, 225–236. [1]

Lange, K., Hunter, D. R., and Yang, I. (2000), "Optimization Transfer Using Surrogate Objective Functions," *Journal of Computational and Graphical Statistics*, 9, 1–59. [4]

Li, Y., and Zhu, J. (2008), "ℓ_1-norm Quantile Regression," *Journal of Computational and Graphical Statistics*, 17, 163–185. [1]

Loh, P. (2017), "Statistical Consistency and Asymptotic Normality for High-Dimensional Robust M-estimators," *The Annals of Statistics*, 45, 866–896. [5]

Pan, X., Sun, Q., and Zhou, W.-X. (2021), "Iteratively Reweighted ℓ_1-penalized Robust Regression," *Electronic Journal of Statistics*, 25 3287–3348. [4]

Pang, L., Lu, W., and Wang, H. J. (2012), "Variance Estimation in Censored Quantile Regression via Induced Smoothing," *Computational Statistics and Data Analysis*, 56, 785–796. [2]

Peng, B., and Wang, L. (2015), "An Iterative Coordinate Descent Algorithm for High-Dimensional Nonconvex Penalized Quantile Regression," *Journal of Computational and Graphical Statistics*, 24, 676–694. [1,3,6]

Petersen, A., and Witten, D. (2019), "Data-Adaptive Additive Modeling," *Statistics in Medicine*, 38, 583–600. [8]

Petersen, A., Witten, D., and Simon, N. (2016), "Fused Lasso Additive Model," *Journal of Computational and Graphical Statistics*, 25, 1005–1025. [8,9]

Portnoy, S., and Koenker, R. (1997), "The Gaussian Hare and the Laplacian Tortoise: Computability of Squared-Error versus Absolute-Error Estimators," *Statistical Science*, 12, 279–300. [3]

Sadhanala, V., and Tibshirani, R. (2019), "Additive Models with Trend Filtering," *The Annals of Statistics*, 47, 3032–3068. [8]

Sherwood, B., and Maidman, A. (2020), *Package* "`rqPen`", version 2.2.2. Reference manual: *https://cran.r-project.org/web/packages/rqPen/rqPen.pdf*. [5]

Simon, N., Friedman, J., Hastie, T., and Tibshirani, R. (2013), "A Sparse-Group Lasso," *Journal of Computational and Graphical Statistics*, 22, 231–245. [1,4,10,11]

Tan, K. M., Wang, L., and Zhou, W.-X. (2022), "High-Dimensional Quantile Regression: Convolution Smoothing and Concave Regularization," *Journal of the Royal Statistical Society*, Series B, 84, 205–233. [2,3,4]

Tibshirani, R. (1996), "Regression Shrinkage and Selection via the Lasso," *Journal of the Royal Statistical Society*, Series B, 58, 267–288. [1,3,4,10]

Tibshirani, R., Saunders, M., Rosset, S., Zhu, J., and Knight, K. (2005), "Sparsity and Smoothness via the Fused Lasso," *Journal of the Royal Statistical Society*, Series B, 67, 91–108. [1]

Tibshirani, R., and Taylor, J. (2011), "The Solution Path of the Generalized Lasso," *The Annals of Statistics*, 39, 1335–1371. [9]

United Nations Development Programme. (2012), *Human Development Indicators*, New York: United Nations Publications. [8]

Wainwright, M. J. (2019), *High-Dimensional Statistics: A Non-Asymptotic Viewpoint*, Cambridge: Cambridge University Press. [1]

Wang, L., Wu, Y., and Li, R. (2012), "Quantile Regression for Analyzing Heterogeneity in Ultra-High Dimension," *Journal of American Statistical Association*, 107, 214–222. [3]

World Bank Group. (2012), *World Development Indicators 2012*, Washington, DC: World Bank Publications. [8]

Wu, J., and Witten, D. (2019), "Flexible and Interpretable Models for Survival Data," *Journal of Computational and Graphical Statistics*, 28, 954–966. [8]

Yi, C., and Huang, J. (2017), "Semismooth Newton Coordinate Descent Algorithm for Elastic-Net Penalized Huber Regression and Quantile

Regression," *Journal of Computational and Graphical Statistics*, 26, 547–557. [1,3]

Yu, L., Lin, N., and Wang, L. (2017), "A Parallel Algorithm for Large-Scale Nonconvex Penalized Quantile Regression," *Journal of Computational and Graphical Statistics*, 26, 935–939. [1,3,11]

Yuan, M., and Lin, Y. (2006), "Model Selection and Estimation in Regression with Grouped Variables," *Journal of the Royal Statistical Society*, Series B, 68, 49–67. [1,4,10,11]

Zhang, C.-H. (2010), "Nearly Unbiased Variable Selection Under Minimax Concave Penalty," *The Annals of Statistics*, 38, 894–942. [1]

Zheng, Q., Peng, L., and He, X. (2015), "Globally Adaptive Quantile Regression with Ultra-High Dimensional Data," *The Annals of Statistics*, 43, 2225–2258. [3]

Zou, H., and Hastie, T. (2005), "Regularization and Variable Selection via the Elastic Net," *Journal of the Royal Statistical Society*, Series B, 67, 301–320. [1,4,10]