Finite Smoothing Algorithm for High-Dimensional Support Vector Machines and Quantile Regression

Qian Tang [*1] **Yikai Zhang** [*1] **Boxiang Wang** [1]

Abstract

This paper introduces a finite smoothing algorithm (FSA), a novel approach to tackle computational challenges in applying support vector machines (SVM) and quantile regression to high-dimensional data. The critical issue with these methods is the non-smooth nature of their loss functions, which traditionally limits the use of highly efficient coordinate descent techniques in high-dimensional settings. FSA innovatively addresses this issue by transforming these loss functions into their smooth counterparts, thereby facilitating more efficient computation. A distinctive feature of FSA is its theoretical foundation: FSA can yield exact solutions, not just approximations, despite the smoothing approach. Our simulation and benchmark tests demonstrate that FSA significantly outpaces its competitors in speed, often by orders of magnitude, while improving or at least maintaining precision. We have implemented FSA in two open-source `R` packages: `hdsvm` for high-dimensional SVM and `hdqr` for high-dimensional quantile regression.

1. Introduction

In the digital epoch, where data reign as the new gold, technological advancements have driven a surge in high-dimensional data, transforming numerous fields such as genetic and genomic research, functional magnetic resonance imaging, clinical trials, and financial market analysis. This burgeoning influx of data has notably fueled the rise of deep learning and artificial intelligence, positioning these methods at the forefront of complex problem-solving. However, their success is typically contingent on the availability of abundant data, a condition that may contrast with the data scarcity encountered in critical areas like medical and psychological research (Alberto et al., 2023). In these domains, ethical and logistical constraints often limit data collection, resulting in datasets with high dimensionality but few observations. In these scenarios, conventional methods like support vector machines (SVM) remain vital due to their robust predictive power with limited data resources. Since its inception, (Cortes & Vapnik, 1995; Vapnik, 1999a;b), SVM has gained considerable popularity for its elegant geometric interpretation and predictive power, at times even overshadowing neural networks. However, its utility is challenged in the realm of high-dimensional data due to its demanding computational requirements. The struggle of SVM underscores an urgent need to enhance its computing efficiency for high-dimensional data analysis.

Furthermore, fields like finance and medicine, where high-dimensional data abound, often place a premium on model interpretability. SVM is prized for its nice geometric interpretation, but a truly effective method for high-dimensional analysis should also excel in selecting important features while discarding irrelevant ones, echoing the scientific hypothesis that only a few important features significantly influence outcomes. The standard SVM without feature selection can suffer from poor classification performance due to noise accumulation (Fan & Fan, 2008). In response, sparse penalized SVMs (Bradley & Mangasarian, 1998; Zhu et al., 2003; Wang et al., 2006; Zhang et al., 2016) were proposed by rephrasing the SVM problem as an ℓ_2 penalized hinge loss (see Hastie et al. (2009), for example) and then replacing the ℓ_2 penalty with sparse penalties like the lasso (Tibshirani, 1996) and elastic net (Zou & Hastie, 2005). This innovation automates feature selection and thus enhances prediction accuracy and interpretability. Further advancements have been attempted with non-convex penalties like SCAD (Fan & Li, 2001) and MCP (Zhang, 2010), renowned for their oracle properties, to further improve prediction accuracy and feature selection. Some recent works acknowledged the generalization error of SVM in high dimensions (Hsu et al., 2021; Ardeshir et al., 2021; Muthukumar et al., 2021) and utilized SVM to interpret transformers in deep learning (Tarzanagh et al., 2023). However, integrating these sparse penalties introduces an additional computational layer to an already intensive SVM process,

[*] Equal contribution [1] Department of Statistics and Actuarial Science, University of Iowa, Iowa City, IA, 52246, United States. Correspondence to: Boxiang Wang <boxiang-wang@uiowa.edu>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

making the development of more efficient algorithms for high-dimensional SVM an imperative pursuit.

Efficient algorithms for solving high-dimensional, sparse penalized methods have long been pivotal in machine learning, statistics, and optimization. The lasso algorithm, initially solved through quadratic programming, was later refined by least angle regression (LARS) (Efron et al., 2004), which exploited its piecewise linearity and expanded its scope to the standard SVM (Hastie et al., 2004) and elastic-net penalized SVM (Wang et al., 2006). However, the true potential of the lasso was not fully recognized until the advent of `glmnet` (Friedman et al., 2010), a highly acclaimed algorithm for its implementation of coordinate descent alongside innovative tricks like the warm start and active set. Coordinate descent is a well-established algorithm in optimization (Wright, 2015; Wu & Lange, 2008; Nesterov, 2012; Tseng & Yun, 2009), and some recent developments are exemplified in Karimireddy et al. (2019); Bertrand & Massias (2021); Nutini et al. (2022). However, the application of coordinate descent to SVM poses specific challenges; in principle, it may fail when the objective is non-differentiable (Luo & Tseng, 1992; Tseng, 2001), as seen in the SVM hinge loss. Such limitations have motivated a spectrum of coordinate descent methods solving SVM through the dual space (Hsieh et al., 2008), or smoothing the SVM loss functions and yielding least squares SVM (Chang et al., 2008), Huberized SVM (Wang et al., 2008), and density convoluted SVM (Wang et al., 2022), for example. Yet, efficiently computing the *exact* solution of SVM tailed on high-dimensional settings remains an open question in the field.

Quantile regression (Koenker & Bassett, 1978; Koenker, 2005; Bassett et al., 2017), like SVM, faces similar challenges, mainly due to its non-differentiable check loss function. Known for its robustness and capacity to comprehensively profile response-feature relationships, quantile regression has found many successful applications in high-dimensional data analysis. Notably, it has been recently integrated into conformal prediction (Shafer & Vovk, 2008; Lei et al., 2018; Tibshirani et al., 2019; Angelopoulos & Bates, 2021), a modern tool for quantifying uncertainties in artificial intelligence systems. The traditional approach to solving quantile regression involves linear programming, but this method falls short in high-dimensional contexts. Various algorithms have been developed for high-dimensional quantile regression; examples include Chen (2007); Peng & Wang (2015); Yi & Huang (2017); Lv et al. (2017); Tan et al. (2022); He et al. (2023). The state-of-the-art solver for high-dimensional quantile regression is `FHDQR` (Gu et al., 2018), combining the ADMM algorithm (Boyd et al., 2011) with coordinate descent and significantly outperforming popular solvers like `quantreg` (Koenker et al., 2018) and `hqreg` (Yi & Huang, 2017).

This work presents a unified algorithm for computing the exact solutions of high-dimensional SVM and quantile regression. Our finite smooth algorithm (FSA) harnesses the power of coordinate descent for high-dimensional computation while overcoming the challenges posed by non-differentiable losses. To achieve this, we propose a smooth approximation for both SVM and quantile regression, and we prove that their exact solutions can be efficiently computed by solving this smooth version and imposing simple linear constraints at non-differentiable points. The smooth versions are solved by a generalized coordinate descent (GCD) algorithm (Yang & Zou, 2013b), which melds standard coordinate descent with the majorization-minimization principle (Hunter & Lange, 2004). In addition, we have expanded the capability of our algorithm to encompass non-convex penalties, such as SCAD and MCP, by incorporating a local linear algorithm (Zou & Li, 2008). Our numerical experiments demonstrate the efficiency and accuracy of FSA in solving high-dimensional SVM and quantile regression challenges. Notably, with the lowest objective values, our algorithm significantly outpaces current state-of-the-art solvers, including `ReHline` (Dai & Qiu, 2023), a newly published algorithm for both SVM and quantile regression, typically by order of magnitude. We have implemented our FSA in two `R` packages, `hdsvm` and `hdqr`, for high-dimensional SVM and quantile regression, respectively. Both packages are published on the Comprehensive R Archive Network (CRAN) [1].

The remainder of this paper is structured as follows: FSA is introduced in Section 2, the GCD algorithm for solving smoothed problems is detailed in Section 3, the efficacy and efficiency of FSA are discussed in Section 4 and Section 5, with technical proofs presented in the appendix.

2. Finite Smoothing Algorithm

In this section, we introduce our unified FSA framework. We begin with high-dimensional SVM, and we then adapt the algorithm to quantile regression.

2.1. Motivation

We first briefly review SVM. Suppose training data consist of n data points, denoted as $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$, where each $\mathbf{x}_i \in \mathbb{R}^p$, and $y_i \in \{-1, 1\}$ for binary classification. Under the high-dimensional setting, we consider $p \gg n$, i.e., the dimension p is much higher than the sample size n.

The linear SVM performs the classification by seeking a hyperplane $\{\mathbf{x} : \beta_0 + \mathbf{x}^\top \boldsymbol{\beta} = 0\}$ that maximizes the mar-

[1] The `R` package `hdsvm` is available at `https://cran.rstudio.com/web/packages/hdsvm/index.html` and the `R` package `hdqr` is at `https://cran.rstudio.com/web/packages/hdqr/index.html`

gin between the two classes and predicts the class label as $\mathrm{sgn}(\beta_0 + \mathbf{x}_{\text{new}}^\top \boldsymbol{\beta})$ for a new data point $\mathbf{x}_{\text{new}}$. SVM can be rephrased as the following loss-plus-penalty form,

$$\min_{\beta_0\in\mathbb{R},\boldsymbol{\beta}\in\mathbb{R}^p} \frac{1}{n}\sum_{i=1}^{n}\left[1 - y_i\left(\beta_0 + \mathbf{x}_i^\top\boldsymbol{\beta}\right)\right]_+ + \frac{\lambda_2}{2}\|\boldsymbol{\beta}\|_2^2, \quad (1)$$

where $(1-u)_+ = \max\{1-u, 0\}$ is the non-differentiable SVM hinge loss.

To apply SVM in high dimensions, we consider a general version of the sparse penalized SVM as follows,

$$(\widehat{\beta}_0, \widehat{\boldsymbol{\beta}}) = \underset{\beta_0\in\mathbb{R},\boldsymbol{\beta}\in\mathbb{R}^p}{\operatorname{argmin}}\ G(\beta_0, \boldsymbol{\beta}), \quad (2)$$

where

$$G(\beta_0, \boldsymbol{\beta}) \equiv \frac{1}{n}\sum_{i=1}^{n}\left[1 - y_i\left(\beta_0 + \mathbf{x}_i^\top\boldsymbol{\beta}\right)\right]_+ + P_{\boldsymbol{\omega},\lambda_1,\lambda_2}(\boldsymbol{\beta}),$$

$$P_{\boldsymbol{\omega},\lambda_1,\lambda_2}(\boldsymbol{\beta}) \equiv \lambda_1\|\boldsymbol{\omega}\circ\boldsymbol{\beta}\|_1 + \frac{\lambda_2}{2}\|\boldsymbol{\beta}\|_2^2.$$

The term $P_{\boldsymbol{\omega},\lambda_1,\lambda_2}(\boldsymbol{\beta})$ is the adaptive elastic-net penalty (Zou & Zhang, 2009), which reduces to the standard SVM when $\lambda_1 = 0$. The weighted ℓ_1 penalty aids in feature selection, and an appropriate selection of the weight $\boldsymbol{\omega}$ leads to the oracle property with nice theory (Zou, 2006). In practice, the tuning parameters λ_1 and λ_2 are often chosen using cross-validation or information criteria.

The main difficulty of solving problem (2) is due to the non-smooth nature of the hinge loss. Common strategies for solving the standard SVM, i.e., problem (1), include resorting to subgradient (Shalev-Shwartz et al., 2007) and transforming the problem into its dual space (Hastie et al., 2009). To achieve better computational efficiency, we consider directly smoothing the non-differentiable objective.

We present a δ-smoothed hinge loss (Wang & Zou, 2022):

$$L_\delta(u) = \begin{cases} 1-u & u \le 1-\delta, \\ \frac{1}{4\delta}[u-(1+\delta)]^2 & 1-\delta < u < 1+\delta, \\ 0 & u \ge 1+\delta. \end{cases}$$

One can show that the above function approaches the hinge loss when δ is small: $0 \le L_\delta(u) - (1-u)_+ \le \delta/4$ for all u. We can thus obtain a smoothed SVM solution by solving a smooth version of problem (2), i.e., $\min_{\beta_0\in\mathbb{R},\boldsymbol{\beta}\in\mathbb{R}^p} G^\delta(\beta_0, \boldsymbol{\beta})$, where

$$G^\delta(\beta_0, \boldsymbol{\beta}) \equiv \frac{1}{n}\sum_{i=1}^{n} L_\delta\left(y_i(\beta_0 + \mathbf{x}_i^\top\boldsymbol{\beta})\right) + P_{\boldsymbol{\omega},\lambda_1,\lambda_2}(\boldsymbol{\beta}).$$

The following proposition quantifies the quality of the smoothed SVM.

Proposition 2.1. *For any $\delta > 0$, it holds that*

$$\min_{\beta_0\in\mathbb{R},\boldsymbol{\beta}\in\mathbb{R}^p} G^\delta(\beta_0, \boldsymbol{\beta}) \le G(\widehat{\beta}_0, \widehat{\boldsymbol{\beta}}) + \delta/4.$$

Although the aforementioned method yields an approximate SVM solution, some distinctive SVM features, for example, the emergence of support vectors, arise from the non-smooth nature of the loss function and may not be preserved in the smoothing approach. Hence, the focus is on advancing this smoothing approach to obtain the exact SVM solution.

2.2. Exact Finite Smoothing Principle

To obtain the exact SVM solution from the smoothing approach, we define a set, $S^\star$, which collects the non-differentiable data points in problem (2),

$$S^\star = \left\{ i : \left| 1 - y_i(\widehat{\beta}_0 + \mathbf{x}_i^\top\widehat{\boldsymbol{\beta}}) \right| = 0 \right\},$$

where $\widehat{\beta}_0$ and $\widehat{\boldsymbol{\beta}}$ are the exact SVM solution in problem (2). It is important to note that if $S^\star$ were known, the exact solution could be attained by adding a linear constraint.

Lemma 2.2. *If $S^\star$ is known, define*

$$(\widehat{\beta}_0^\delta, \widehat{\boldsymbol{\beta}}^\delta) = \underset{\beta_0\in\mathbb{R},\boldsymbol{\beta}\in\mathbb{R}^p}{\operatorname{argmin}}\ G^\delta(\beta_0, \boldsymbol{\beta}),$$
$$\textit{subject to}\ \ 1 = y_i(\beta_0 + \mathbf{x}_i^\top\boldsymbol{\beta}),\ i \in S^\star,$$

then $(\widehat{\beta}_0, \widehat{\boldsymbol{\beta}}) = (\widehat{\beta}_0^\delta, \widehat{\boldsymbol{\beta}}^\delta)$ holds.

Lemma 2.2 offers a method to derive the exact SVM solution through solving a smoothed problem, but its practical application is limited due to the unknown nature of set $S^\star$ prior to obtaining $(\widehat{\beta}_0, \widehat{\boldsymbol{\beta}})$. In response, we introduce a modified version of Lemma 2.2. This adaptation highlights that determining a subset, $\widehat{S}^\delta$, of $S^\star$ is adequate for accurately obtaining the exact SVM solution. We define some quantities: let $\delta_0 = \min_{i\notin S^\star}\{|1 - y_i(\widehat{\beta}_0 + \mathbf{x}_i^\top\widehat{\boldsymbol{\beta}})|\} > 0$, $C_{\delta_0/2} = \{(\beta_0, \boldsymbol{\beta}) : \|\beta_0\mathbf{1}_n + \mathbf{x}^\top\boldsymbol{\beta} - \widehat{\beta}_0\mathbf{1}_n - \mathbf{x}^\top\widehat{\boldsymbol{\beta}}\|_\infty \geqslant \delta_0/2\}$, $\eta = \inf_{(\beta_0,\boldsymbol{\beta})\in C_{\delta_0/2}}\{G(\beta_0, \boldsymbol{\beta}) - G(\widehat{\beta}_0, \widehat{\boldsymbol{\beta}})\}$, and $\delta^\sharp = \min\{\delta_0/2, 4\eta\}$. We present the following theorem.

Theorem 2.3. *For any $\delta \in (0, \delta^\sharp)$, there exists a set $\widehat{S}^\delta \subseteq S^\star$ such that if we solve*

$$(\widehat{\beta}_0^\delta, \widehat{\boldsymbol{\beta}}^\delta) = \underset{\beta_0\in\mathbb{R},\boldsymbol{\beta}\in\mathbb{R}^p}{\operatorname{argmin}}\ G^\delta(\beta_0, \boldsymbol{\beta}),$$
$$\textit{subject to}\ \ 1 = y_i(\beta_0 + \mathbf{x}_i^\top\boldsymbol{\beta}),\ i \in \widehat{S}^\delta, \quad (3)$$

and define $\widetilde{S}^\delta = \{i\colon -\delta \le 1 - y_i(\widehat{\beta}_0^\delta + \mathbf{x}_i^\top\widehat{\boldsymbol{\beta}}^\delta) \le \delta\}$ accordingly, then we have $\widetilde{S}^\delta = \widehat{S}^\delta$. Further, we have $(\widehat{\beta}_0, \widehat{\boldsymbol{\beta}}) = (\widehat{\beta}_0^\delta, \widehat{\boldsymbol{\beta}}^\delta)$ for any $\delta \in (0, \delta^\sharp)$.

Algorithm 1 `hdsvm`

```
Input: $\mathbf{x}_i, y_i$.
Initialize: $\delta \leftarrow 1$.
repeat
  repeat
    Initialize $\widetilde{S} = \emptyset$.
    Solve problem (4) using GCD in Section 3.
    Update $\widetilde{S}$ by equation (5).
  until the set $\widetilde{S}$ converges (in finite iterates).
  Update $\delta \leftarrow \delta/4$.
until the KKT condition of problem (2) is satisfied.
```

Theorem 2.3 relaxes the unrealistic condition in Lemma 2.2 by questing for a practically achievable set $\widehat{S}^\delta$ instead of $S^\star$. Theorem 2.4 further outlines a method for the explicit construction of $\widehat{S}^\delta$. When Theorem 2.4 is employed to any initial subset $\widetilde{S}$ within $S^\star$, $\widetilde{S}^\delta$ is yielded, augmenting $\widetilde{S}$ and progressively approximating $S^\star$. Due to the finite sample size, through successive iterations, $\widehat{S}^\delta$ can be eventually constructed within a finite number of iterates.

Theorem 2.4. *For any set $\widetilde{S} \subseteq S^\star$ and $\delta \in (0, \delta^\sharp)$, define*

$$(\widetilde{\beta}_0^\delta, \widetilde{\boldsymbol{\beta}}^\delta) = \operatorname*{argmin}_{\beta_0\in\mathbb{R},\boldsymbol{\beta}\in\mathbb{R}^p} G^\delta(\beta_0, \boldsymbol{\beta}), \quad \text{subject to } \; 1 = y_i(\beta_0 + \mathbf{x}_i^\top \boldsymbol{\beta}),\ i \in \widetilde{S}, \tag{4}$$

and let

$$\widetilde{S}^\delta = \{i \colon -\delta \leq 1 - y_i(\widetilde{\beta}_0^\delta + \mathbf{x}_i^\top \widetilde{\boldsymbol{\beta}}^\delta) \leq \delta\}, \tag{5}$$

then the following holds: $\widetilde{S} \subseteq \widetilde{S}^\delta \subseteq S^\star$.

In practice, $\delta^\sharp$ is also unknown. To address this challenge, we have developed a procedure that involves executing the previously mentioned steps repeatedly, each time with a sequentially reduced δ value. The algorithm concludes once we find a solution that satisfies the Karush–Kuhn–Tucker (KKT) condition of the exact SVM, i.e., problem (1). In practice, we start this sequence with $\delta = 1$ and then decrease it in each iteration to a quarter of its former value, that is, $\delta \leftarrow \delta/4$. The algorithm is summarized in Algorithm 1.

2.3. Quantile Regression

The FSA framework can be naturally adapted to high-dimensional quantile regression, solving $(\widehat{\beta}_0^{\mathrm{qr}}, \widehat{\boldsymbol{\beta}}^{\mathrm{qr}})$ from

$$\min_{\beta_0,\boldsymbol{\beta}} \sum_{i=1}^n \rho_\tau(y_i - \beta_0 - \boldsymbol{x}_i^\top \boldsymbol{\beta}) + \lambda_1 \|\boldsymbol{\omega} \circ \boldsymbol{\beta}\|_1 + \frac{\lambda_2}{2}\|\boldsymbol{\beta}\|_2^2, \tag{6}$$

where $\tau \in (0,1)$ is a given quantile level and $\rho_\tau(t) = t(\tau - I(t < 0))$, for $t \in \mathbb{R}$, is the quantile loss, or namely, the *check loss*. To address the non-smooth nature of the check loss function, we introduce a δ-smoothed check loss:

$$H_{\delta,\tau}(t) = \begin{cases} (\tau - 1)t & \text{if } \; t < -\delta, \\ \frac{t^2}{4\delta} + t(\tau - \frac{1}{2}) + \frac{\delta}{4} & \text{if } \; -\delta \leq t \leq \delta, \\ \tau t & \text{if } \; t > \delta. \end{cases}$$

Similar to SVM, one can show the exact solution of problem (6) can be obtained from the following constrained optimization problem:

$$\begin{aligned} &\min_{\beta_0,\boldsymbol{\beta}} \frac{1}{n}\sum_{i=1}^n H_{\delta,\tau}(y_i - \beta_0 - \mathbf{x}_i^\top \boldsymbol{\beta}) + \lambda_1 \|\boldsymbol{\omega} \circ \boldsymbol{\beta}\|_1 + \frac{\lambda_2}{2}\|\boldsymbol{\beta}\|_2^2, \\ &\quad \text{subject to } y_i = \beta_0 + \mathbf{x}_i^\top \boldsymbol{\beta},\ \forall i \in E^\star, \\ &\quad \text{where } E^\star = \left\{ i : \left| y_i - \widehat{\beta}_0^{\mathrm{qr}} - \mathbf{x}_i^\top \widehat{\boldsymbol{\beta}}^{\mathrm{qr}} \right| = 0 \right\}. \end{aligned} \tag{7}$$

3. Generalized Coordinate Descent Algorithm

In this section, we develop an efficient algorithm designed to compute the solution path for problem (4). To solve problem (4), we employ the augmented Lagrangian method:

$$\min_{\beta_0\in\mathbb{R},\boldsymbol{\beta}\in\mathbb{R}^p,\boldsymbol{\theta}\in\mathbb{R}^{|\widetilde{S}|}} \mathcal{L}_\sigma(\beta_0, \boldsymbol{\beta}, \boldsymbol{\theta}),$$

where σ is a constant, $\boldsymbol{\theta} \in \mathbb{R}^{|\widetilde{S}|}$ is the Lagrangian multiplier, $|\widetilde{S}|$ denote the number of elements in the set $\widetilde{S}$, and

$$\begin{aligned} &\mathcal{L}_\sigma(\beta_0, \boldsymbol{\beta}, \boldsymbol{\theta}) \\ \equiv &\frac{1}{n}\sum_{i=1}^n L_\delta(y_i\left(\beta_0 + \mathbf{x}_i^\top \boldsymbol{\beta}\right)) + \lambda_1 \|\boldsymbol{\omega} \circ \boldsymbol{\beta}\|_1 + \frac{\lambda_2}{2}\|\boldsymbol{\beta}\|_2^2 \\ &+ \sum_{i\in\widetilde{S}} \theta_i (1 - y_i(\beta_0 + \mathbf{x}_i^\top \boldsymbol{\beta})) \\ &+ \frac{\sigma}{2}\sum_{i\in\widetilde{S}} (1 - y_i(\beta_0 + \mathbf{x}_i^\top \boldsymbol{\beta}))^2. \end{aligned} \tag{8}$$

In a coordinate-wise manner, suppose $\beta_1, \beta_2, \dots, \beta_{j-1}$ have been updated and we now update β_j. Let $(\widetilde{\beta}_0, \widetilde{\boldsymbol{\beta}})$ be the current solution and $r_i = y_i(\widetilde{\beta}_0 + \mathbf{x}_i^\top \widetilde{\boldsymbol{\beta}})$. To update β_j, we define the coordinate-wise update function:

$$\begin{aligned} F(\beta_j | \widetilde{\beta}_0, \widetilde{\boldsymbol{\beta}}) \equiv &\frac{1}{n}\sum_{i=1}^n L_\delta(r_i + y_i x_{ij}(\beta_j - \widetilde{\beta}_j)) + \lambda_1 \omega_j |\beta_j| \\ &+ \frac{\lambda_2}{2}\beta_j^2 - \sum_{i\in\widetilde{S}} \theta_i y_i x_{ij} \beta_j + \sigma \sum_{i\in\widetilde{S}} \sum_{t\neq j} \widetilde{\beta}_t x_{it} x_{ij} \beta_j \\ &- \sigma \sum_{i\in\widetilde{S}} y_i x_{ij} \beta_j + \frac{\sigma}{2}\sum_{i\in\widetilde{S}} (x_{ij}\beta_j)^2 + \sigma \sum_{i\in\widetilde{S}} x_{ij} \beta_j \widetilde{\beta}_0. \end{aligned}$$

Then the standard coordinate descent algorithm suggests cyclically minimizing $F(\beta_j \mid \widetilde{\beta}_0, \widetilde{\boldsymbol{\beta}})$ in terms of β_j, but this problem does not have a closed-form solution. To handle this, we consider a generalized coordinate descent (GCD) algorithm (Yang & Zou, 2013a) based on the majorization-minimization principle (Hunter & Lange, 2000).

Specifically, we define a majorization function

$$\begin{aligned}
&Q\left(\beta_j|\widetilde{\beta}_0,\widetilde{\boldsymbol{\beta}}\right)=\frac{1}{n}\sum_{i=1}^{n}L_\delta\left(r_i\right)+\frac{1}{n}\sum_{i=1}^{n}L'_\delta\left(r_i\right)y_ix_{ij}\left(\beta_j-\widetilde{\beta}_j\right)\\
&+\frac{1}{4\delta}\left(\beta_j-\widetilde{\beta}_j\right)^2+\lambda_1\omega_j|\beta_j|+\frac{\lambda_2}{2}\beta_j^2-\sum_{i\in\widetilde{S}}\theta_iy_ix_{ij}\beta_j\\
&-\sigma\sum_{i\in\widetilde{S}}y_ix_{ij}\beta_j+\sigma\sum_{i\in\widetilde{S}}\sum_{t\neq j}\widetilde{\beta}_tx_{it}x_{ij}\beta_j\\
&+\frac{\sigma}{2}\sum_{i\in\widetilde{S}}(x_{ij}\beta_j)^2+\sigma\sum_{i\in\widetilde{S}}x_{ij}\beta_j\widetilde{\beta}_0.
\end{aligned}$$

It holds that $F(\beta_j|\widetilde{\beta}_0,\widetilde{\boldsymbol{\beta}})\leq Q(\beta_j|\widetilde{\beta}_0,\widetilde{\boldsymbol{\beta}})$ for all β_j and the equality holds only when $\beta_j=\widetilde{\beta}_j$. This is because the derivative of $L_\delta(\cdot)$ is Lipschitz continuous.

We can efficiently minimize $Q(\beta_j|\widetilde{\beta}_0,\widetilde{\boldsymbol{\beta}})$ by a simple soft-thresholding operator:

$$\begin{aligned}
\beta_j^{\text{new}}&=\underset{\beta_j}{\operatorname{argmin}}\;Q(\beta_j\mid\widetilde{\beta}_0,\widetilde{\boldsymbol{\beta}})\\
&=S\Bigg(\frac{1}{2\delta}\widetilde{\beta}_j+\sum_{i\in\widetilde{S}}x_{ij}(\theta_iy_i+\sigma(y_i-\sum_{t\neq j}x_{it}\widetilde{\beta}_t-\widetilde{\beta}_0))\\
&-\frac{1}{n}\sum_{i=1}^{n}L'_\delta(r_i)y_ix_{ij},\lambda_1\omega_j\Bigg)/(\frac{1}{2\delta}+\lambda_2+\sigma\sum_{i\in\widetilde{S}}x_{ij}^2),
\end{aligned}\tag{9}$$

where $S(z,t)=(|z|-t)_+\operatorname{sgn}(z)$. We set $\widetilde{\beta}_j=\widetilde{\beta}_j^{\text{new}}$ and proceed to the next coordinate.

After updating all $\beta_j, j=1,2,\ldots,p$, with $c=\sigma|\widetilde{S}|+\frac{1}{2\delta}$, we update the intercept β_0^{new} as follows:

$$\widetilde{\beta}_0-\frac{1}{c}\left(\frac{1}{n}\sum_{i=1}^{n}L'_\delta\left(r_i\right)y_i+\sum_{i\in\widetilde{S}}\sigma(y_i-\boldsymbol{x}_i^\top\widetilde{\boldsymbol{\beta}}-\widetilde{\beta}_0)+\theta_iy_i\right).\tag{10}$$

We then update the Lagrangian multiplier as

$$\theta_i^{\text{new}}=\theta_i-\sigma\left(1-y_i(\mathbf{x}_i^\top\widetilde{\boldsymbol{\beta}}+\widetilde{\beta}_0)\right),$$

for each $i\in\widetilde{S}$. The above steps are repeated until the convergence. The GCD algorithm for SVM is summarized in Algorithm 2 in the appendix.

Consequently, we have built the GCD algorithm on the augmented Lagrangian method to solve problem (4). We then present the following theorem to show the linear convergence of the algorithm. Similar techniques have been used in the literature, for example, Boyd et al. (2011); Gu et al. (2018); He & Yuan (2012).

Algorithm 2 The GCD algorithm for high-dimensional SVM

```
1. Initialize $(\widetilde{\beta}_0,\widetilde{\boldsymbol{\beta}})$.
2. Cyclic coordinate descent, for $j=1,2,\ldots,p$:
    (a) Compute $r_i=y_i(\widetilde{\beta}_0+\mathbf{x}_i^\top\widetilde{\boldsymbol{\beta}})$.
    (b) Compute $\widetilde{\beta}_j^{\text{new}}$ using the update formula (9).
    (c) Set $\widetilde{\beta}_j=\widetilde{\beta}_j^{\text{new}}$.
3. Update the intercept term:
    (a) Compute $r_i=y_i(\widetilde{\beta}_0+\mathbf{x}_i^\top\widetilde{\boldsymbol{\beta}})$.
    (b) Compute $\widetilde{\beta}_0^{\text{new}}$ using the update formula (10).
    (c) Set $\widetilde{\beta}_0=\widetilde{\beta}_0^{\text{new}}$.
4. Update $\boldsymbol{\theta}$, for all $i\in\widetilde{S}$:
    (a) Update $\widetilde{\theta}_i^{\text{new}}=\widetilde{\theta}_i-\sigma\left(1-y_i(\widetilde{\beta}_0+\mathbf{x}_i^\top\widetilde{\boldsymbol{\beta}})\right)$.
    (b) Set $\widetilde{\theta}_i=\widetilde{\theta}_i^{\text{new}}$.
5. Repeat steps 2-4 until convergence of $(\widetilde{\beta}_0,\widetilde{\boldsymbol{\beta}})$.
```

Theorem 3.1. *Algorithm 2 ensures the linear convergence of the objective $G^\delta(\widetilde{\beta}_0,\widetilde{\boldsymbol{\beta}})$ to the optimal value of problem (4).*

Since FSA is guaranteed to converge in finite iterates, the whole procedure for solving high-dimensional SVM in Algorithm 1 converges linearly.

A similar GCD algorithm can also be developed for quantile regression. The details of the GCD algorithm are given in Algorithm 3 in the appendix.

Implementation In our implementation, to enhance the computational efficiency for the entire procedure, including the use of cross-validation to select the tuning parameters, we employ the warm-start and active-set strategies to compute the solution path as λ_1 varies.

In particular, we compute the solution path $(\widehat{\beta}_0^{[k]}),(\widehat{\boldsymbol{\beta}}^{[k]})$ for a sequence of decreasing λ_1 values, $\lambda_1^{[1]}>\lambda_1^{[2]}>\ldots>\lambda_1^{[K]}$. If $\lambda_1^{[k]}>\frac{1}{n}\max_j|\sum_{i=1}^n L'_\delta(\widehat{\beta}_0)y_ix_{ij})/w_j|$, the KKT condition implies that all $\beta_j=0$; otherwise, we employ the *warm-start* strategy to use the solution at $\lambda_1^{[k-1]}$ as the initial value for computing the solution at $\lambda_1^{[k]}$.

We also use the *active-set* idea to compute the solution at each λ_1. The active set contains those variables whose current coefficients are nonzero. After a complete cycle through all the variables, we only apply coordinate descent on the active set until the convergence. We then run another complete cycle to see if the active set changes; otherwise, the algorithm stops.

In addition, the safe rule (Ghaoui et al., 2010; Wang & Yang, 2022) or the strong rule (Tibshirani et al., 2012) can be used to further accelerate our algorithm. We take the strong rule as an example: with the optimizer $(\widetilde{\beta}_0^{[k]}, \widetilde{\boldsymbol{\beta}}^{[k]})$ determined in $\lambda_1^{[k]}$, the strong rule is applied to guess whether $\boldsymbol{\beta}^{[k+1]} = 0$ in subsequent $\lambda_1^{[k+1]}$. Specifically, if some j satisfies that

$$\left|\frac{1}{n}\sum_{i=1}^{n} L'_\delta\left(y_i\left(\widetilde{\beta}_0^{[k]}+\mathbf{x}_i^\top\widetilde{\boldsymbol{\beta}}^{[k]}\right)\right)y_i\mathbf{x}_{ij}\right| \geq 2\lambda_1^{[k+1]}-\lambda_1^{[k]},$$

then $\beta_j^{[k+1]} = 0$, and its computation can be saved in the subsequent coordinate descent step. In practice, it is common to verify the KKT condition to confirm that all the variables are correctly eliminated by the strong rule.

Non-convex penalties We now demonstrate the capacity of our algorithm to handle non-convex penalties such as SCAD (Fan & Li, 2001) and MCP (Zhang, 2010). The non-convex penalties can be imposed on SVM in place of the weighted ℓ_1 penalty in problem (2). To the high-dimensional SVM with non-convex penalties, we apply the local linear algorithm (LLA) (Zou & Li, 2008). Specifically, with some initial solution $(\widehat{\beta}_0^{(k)}, \widehat{\boldsymbol{\beta}}^{(k)})$ and p'_λ being the derivative of the penalty, we solve problem (2) with the weight $\omega_j = p'_\lambda(|\widehat{\beta}_j^{(k)}|)$ to obtain $(\widehat{\beta}_0^{(k+1)}, \widehat{\boldsymbol{\beta}}^{(k+1)})$, which is used as the initial solution for the next iterate. Therefore, the GCD algorithm can be naturally extended to handle non-convex penalties.

We have implemented our algorithm for solving high-dimensional SVM in an R package hdsvm and the algorithm for quantile regression in an R package hdqr.

4. Simulation

In this section, we demonstrate the quality and efficiency of FSA for both high-dimensional SVM and quantile regression using extensive simulation data.

4.1. SVM

We first showcase the performance of high-dimensional SVM. In this section, the response variables are binary, and the dimension p is 3000 or 10000. In each example, training data contain 200 data points, 100 of which are from the positive class and the other 100 from the negative class.

Table 1. High-dimensional SVM: comparison of the objective values of problem (2) and run time (in seconds) for simulation data. All the quantities are averaged over 20 independent runs.

EXAMPLE		HDSVM	REHLINE	CVXR	DRSVM
$\rho = 0.2$					
$p = 3000$	OBJ	0.74	0.74	0.74	0.82
	TIME	0.04	88.94	211.01	454.93
$p = 10000$	OBJ	0.78	0.79	0.81	0.86
	TIME	0.12	1389.28	1425.43	1260.61
$\rho = 0.7$					
$p = 3000$	OBJ	0.78	0.79	0.79	0.89
	TIME	0.03	49.01	127.18	350.53
$p = 10000$	OBJ	0.82	0.83	0.85	0.90
	TIME	0.10	1004.19	1588.42	1665.29
$\rho = 0.9$					
$p = 3000$	OBJ	0.80	0.81	0.81	0.94
	TIME	0.03	78.48	222.35	419.07
$p = 10000$	OBJ	0.83	0.84	0.86	0.93
	TIME	0.10	966.58	1387.79	1869.43

Simulation data are generated following Wang et al. (2006). The positive class has a normal distribution with mean $\boldsymbol{\mu}_+$ and covariance $\boldsymbol{\Sigma}$, where $\boldsymbol{\mu}_+ = 0.7$ for the first five features and 0 in others,

$$\boldsymbol{\Sigma} = \begin{pmatrix} \boldsymbol{\Sigma}^\star_{5\times 5} & \mathbf{0}_{5\times(p-5)} \\ \mathbf{0}_{(p-5)\times 5} & \mathbf{I}_{(p-5)\times(p-5)} \end{pmatrix},$$

and the (i,j)th element of $\Sigma^\star$ equals $\rho^{|i-j|}, \rho \in \{0.2, 0.7, 0.9\}$. The negative class has the same distribution except for a different mean $\boldsymbol{\mu}_- = -\boldsymbol{\mu}_+$.

In each example, we fit hdsvm algorithm with the tuning parameters selected by cross-validation. With the chosen tuning parameters, we fit DrSVM (Wang et al., 2006), ReHline (Dai & Qiu, 2023), and CVXR (Fu et al., 2020), and compare the objective value of problem (2) and run time. As shown in Table 1, our hdsvm is about four to five orders of magnitude faster than all the other solvers, and the objective values are consistently the lowest.

Using the examples where $p = 10,000$, we further illustrate the effectiveness of our algorithm in addressing non-convex penalties. In each case, we applied a high-dimensional SVM with both SCAD and MCP penalties across 20 independent replicates. The algorithm was assessed by evaluating the test error across $2,000$ independently generated data points, quantifying the number of accurately identified features (true positives), and determining the number of erroneously selected features (false positives), in addition to measuring the run time. As shown in Table 2, both SCAD and MCP exhibit a slight improvement in test error compared to the elastic net. Notably, all three approaches consistently

Table 2. Comparison of test error (denoted by ERR), the number of correctly selected features (denoted by C), incorrectly selected features (denoted by IC), and run time between SVM with elastic-net and MCP penalties. All the quantities are averaged over 20 independent runs and the standard errors are given in parentheses.

EXAMPLE		HDSVM	MCP	SCAD
$\rho = 0.2$	ERR	0.11(0.02)	0.10(0.01)	0.11(0.02)
	C	5(0)	5(0)	5(0)
	IC	7(12)	1(1)	5(10)
	TIME	0.12	0.20	0.23
$\rho = 0.7$	ERR	0.21(0.03)	0.20(0.01)	0.21(0.03)
	C	5(0)	5(0)	5(0)
	IC	10(15)	1(2)	5(10)
	TIME	0.10	0.21	0.27
$\rho = 0.9$	ERR	0.25(0.03)	0.24(0.02)	0.25(0.03)
	C	5(0)	5(0)	5(0)
	IC	13(20)	3(8)	5(11)
	TIME	0.10	0.22	0.26

Table 3. High-dimensional quantile regression: comparison of the objective values of problem (6) and run time (in seconds) for the simulation data with $n = 200, p = 3,000$. All the quantities are averaged over 20 independent runs.

τ		HDQR	FHDQR	REHLINE	CVXR
0.1	OBJ	0.64	0.66	0.78	0.65
	TIME	0.03	1.43	96.98	421.91
0.3	OBJ	1.24	1.25	1.41	1.25
	TIME	0.04	1.50	102.67	381.61
0.5	OBJ	0.87	0.87	0.91	0.88
	TIME	0.38	2.02	55.81	180.48
0.7	OBJ	1.28	1.29	1.45	1.29
	TIME	0.03	0.84	102.06	344.29
0.9	OBJ	0.64	0.66	0.81	0.65
	TIME	0.02	2.28	137.32	496.06

identified the five key features, with SCAD and MCP introducing significantly fewer irrelevant features. Moreover, the integration of non-convex penalties through our algorithm does not significantly increase computational overhead; the runtime is merely about double that of the elastic net.

4.2. Quantile Regression

We compare our `hdqr` algorithms with `FHDQR`, `ReHline`, and `CVXR` in `R`. We considered five different quantile levels: $\tau = 0.1, 0.3$, 0.5, 0.7, and 0.9. We considered a popular simulation model from Friedman et al. (2010). We generated Gaussian data with $n = 200$ observations and $p = 3,000$, or $10,000$ features, where each pair of features has an identical correlation $\rho = 0.1$. The response values were generated by $y = \sum_{j=1}^{\rho} x_j\beta_j + k \cdot z$, where $\beta_j = (-1)^j \exp(-2(j-1)/20)$, $z \sim \mathrm{N}(0,1)$, and k is chosen so that the signal-to-noise ratio is 3.0.

Table 4. High-dimensional quantile regression: comparison of the objective values of problem (6) and run time (in seconds) for the simulation data with $n = 200, p = 10,000$. All the quantities are averaged over 20 independent runs.

τ		HDQR	FHDQR	REHLINE	CVXR
0.1	OBJ	0.65	0.69	1.10	0.76
	TIME	0.08	3.99	949.23	953.62
0.3	OBJ	1.25	1.28	1.86	1.44
	TIME	0.08	2.91	1273.19	1295.89
0.5	OBJ	0.80	0.81	1.06	0.88
	TIME	1.40	6.42	479.97	566.37
0.7	OBJ	1.27	1.29	1.82	1.42
	TIME	0.10	5.75	1336.40	1351.86
0.9	OBJ	0.63	0.66	1.10	0.72
	TIME	0.08	5.55	1144.30	1537.25

Tables 3 and 4 showcase the average computation time and objective values of problem (6) at the optimal tuning parameters chosen by cross-validation. Our algorithm, `hdqr`, consistently outperforms the other solvers in speed across all examples. The computational speed of `ReHline` and `CVXR` tends to decrease significantly as the parameter p increases. It is noteworthy that `hdqr` achieves the lowest objective values, which are significantly smaller than the other three solvers.

4.3. Path Solution and Scalability

We compare the run time for computing the solution path across the entire range of hyperparameters. Specifically, for every solver, we computed the solution path with 50 different pairs of (λ_1, λ_2), where $\lambda_2 = 100, 10, 1, 0.1, 0.01$ and λ_1 is chosen from 10 values that are uniformly distributed on the log scale between 10 and 0.001. The results for high-dimensional SVM and quantile regression are detailed in Tables 5, 6, and 7, highlighting our algorithms' significant advantages.

We repeated the numerical studies in previous sections, except for different combinations of (n, p). We first fixed $p = 3,000$ and varied $n = 200, 400, 600, 800$, and 1000. We further investigated the scalability for both n and p. With a fixed ratio $p/n = 15$, varied $n = 200, 400, 600, 800$, and 1000. Tables 8 and 9 show the computation times for fitting high-dimensional SVM and quantile regression models, respectively, demonstrating our algorithm's linear scalability with increasing samples and dimensions.

Table 5. High-dimensional SVM: comparison of the objective values of problem (2) and run time (in seconds) for simulation data. All the quantities are averaged over 20 independent runs.

EXAMPLE	HDSVM	REHLINE	CVXR	DRSVM
$\rho = 0.2$				
$p = 3000$	6.81	770.48	7463.17	10343.14
$p = 10000$	25.21	13767.63	41543.43	51973.17
$\rho = 0.7$				
$p = 3000$	7.99	783.52	7390.09	12662.12
$p = 10000$	26.81	16918.78	53069.30	71533.04
$\rho = 0.9$				
$p = 3000$	8.01	780.88	7747.43	11514.14
$p = 10000$	44.57	28513.21	88306.99	119057.20

Table 6. High-dimensional quantile regression: comparison of computation times (in seconds) for problem (6) using simulation data ($n = 200, p = 3000$) across different settings of the hyperparameters (λ_1, λ_2). The presented values represent averages computed over 20 independent runs.

$n = 200$	HDQR	FHDQR	REHLINE	CVXR
$\tau = 0.1$	7.42	22.54	1646.84	6862.58
$\tau = 0.3$	5.92	17.80	1931.15	7860.69
$\tau = 0.5$	5.65	17.02	2015.02	10118.75
$\tau = 0.7$	5.83	17.35	1990.67	9547.73
$\tau = 0.9$	7.01	23.01	1935.57	7525.73

Table 7. High-dimensional quantile regression: comparison of computation times (in seconds) for problem (6) using the microarray data ($n = 120, p = 3000$) across different settings of the hyperparameters (λ_1, λ_2). The presented values represent averages computed over 20 independent runs.

$n = 120$	HDQR	FHDQR	REHLINE	CVXR
$\tau = 0.1$	0.74	11.59	4147.52	3349.96
$\tau = 0.3$	0.76	5.46	4480.60	4474.42
$\tau = 0.5$	0.68	2.67	4722.68	4353.30
$\tau = 0.7$	0.63	1.81	4389.44	3225.64
$\tau = 0.9$	0.81	3.21	4509.93	3590.54

Table 8. High-dimensional SVM: comparison of the objective values of problem (2) and run time (in seconds) for simulation data with different experimental sizes. All the quantities are averaged over 20 independent runs.

$p = 3000$					
n	200	400	600	800	1000
$\rho = 0.2$	7.47	15.76	25.19	34.77	47.13
$\rho = 0.7$	9.75	25.80	29.76	38.11	49.98
$\rho = 0.9$	8.94	18.89	29.55	40.75	49.74
$p/n = 15$					
n	200	400	600	800	1000
$\rho = 0.2$	7.47	28.03	55.29	87.83	125.67
$\rho = 0.7$	9.75	31.32	57.68	97.87	138.59
$\rho = 0.9$	8.94	32.27	61.50	122.29	162.01

5. Benchmark Data Applications

In this section, we first evaluate the performance of `hdsvm` on five benchmark high-dimensional data (Mai & Zou, 2015; Sorace & Zhan, 2003; Graham et al., 2010; Alon et al., 1999; Golub et al., 1999), with the dimension varying between $2,000$ and $22,283$. Table 10 exhibits the objective value of problem (2) and run time. We see that our `hdsvm` consistently delivers better performance than the other solvers in terms of both computational efficiency and precision.

We then evaluate the performance of `hdqr` using a data set reported in Scheetz et al. (2006). This benchmark data set encompasses gene expression levels across over $31,000$ probes in 120 twelve-week-old laboratory rats, aimed at exploring gene regulation in mammalian eyes and contributing to the understanding of genetic variations impacting human eyesight. With penalized quantile regression, we focus on $18,976$ probes identified for sufficient variability following the criteria of Scheetz et al. (2006) and Huang et al. (2008). Notably, the probe `1389163\_at`, linked to the `TRIM32` gene and associated with Bardet–Biedl Syndrome (Chiang et al., 2006), is of particular interest. Our analysis examines the dependency of TRIM32 expression on the other 18,975 genes. After standardizing these gene expressions, we select the top p probes based on variance, considering two scenarios: $p = 3,000$ and $p = 10,000$.

In our study, we compare the objective values and computation time using the solvers `hdqr`, `FHDQR`, `ReHline`, and `CVXR`, all applied to high-dimensional quantile regression at quantile levels $\tau = 0.1, 0.3, 0.5, 0.7,$ and 0.9. As detailed in Tables 11 and 12, `hdqr` and `FHDQR` clearly outperform the other two solvers in terms of computational speed, with `hdqr` emerging as the fastest. As the dimension grows, the efficiency advantage of `hdqr` becomes more marked. Specifically, `hdqr` is ten times faster than `FHDQR` and at least $3,000$ times faster than `ReHline` and `CVXR`. Notably, at $p = 10,000$ and $\tau = 0.1$, the difference in com-

Table 9. High-dimensional QR: comparison of run time (in seconds) of problem (6) for simulation data with different experimental sizes. All the quantities are averaged over 20 independent runs.

$p = 3000$					
n	200	400	600	800	1000
$\tau = 0.1$	9.71	18.56	25.26	33.40	35.28
$\tau = 0.3$	8.01	15.49	22.13	29.95	34.41
$\tau = 0.5$	5.89	9.20	18.44	18.27	25.99
$\tau = 0.7$	6.01	9.88	16.19	21.37	25.74
$\tau = 0.9$	6.90	13.81	19.12	24.56	29.61
$p/n = 15$					
n	200	400	600	800	1000
$\tau = 0.1$	9.71	31.31	60.27	109.91	174.18
$\tau = 0.3$	8.01	25.93	48.85	83.38	118.38
$\tau = 0.5$	5.89	16.66	37.02	50.58	69.92
$\tau = 0.7$	6.01	23.43	39.88	57.64	94.21
$\tau = 0.9$	6.90	28.30	68.05	77.62	126.17

Table 10. High-dimensional SVM: comparison of the objective values of problem (2) and run time (in seconds) for five benchmark high-dimension data from the UCI machine learning repository. All the quantities are averaged over 20 independent runs.

DATA		HDSVM	REHLINE	CVXR	DRSVM
BREAST					
$n = 42$	OBJ	0.45	0.56	0.48	0.50
$p = 22283$	TIME	0.27	3979.62	234.55	59.71
COLON					
$n = 62$	OBJ	0.48	0.51	0.48	0.52
$p = 2000$	TIME	0.07	13.88	13.33	8.11
LEUK					
$n = 72$	OBJ	0.17	0.20	0.22	0.22
$p = 7128$	TIME	0.69	96.13	39.27	40.54
MALARIA					
$n = 71$	OBJ	0.01	0.01	0.01	0.06
$p = 22283$	TIME	24.53	275.00	85.70	112.50
OVARIAN					
$n = 253$	OBJ	0.03	0.04	0.03	0.14
$p = 15154$	TIME	2.88	1880.38	214.06	1230.38

putation time between hdqr and FHDQR diminishes, while FHDQR's objective value is large. ReHline's objective values are notably lower than those of the other solvers.

Table 11. High-dimensional quantile regression: comparison of the objective values of problem (6) and run time (in seconds) for the microarray data ($n = 120, p = 3,000$) in Scheetz et al. (2006). All the quantities are averaged over 20 independent runs.

τ		HDQR	FHDQR	REHLINE	CVXR
0.1	OBJ	0.02	0.02	0.06	0.02
	TIME	0.02	1.21	127.77	121.98
0.3	OBJ	0.03	0.03	0.07	0.04
	TIME	0.03	0.46	130.71	141.29
0.5	OBJ	0.03	0.03	0.07	0.04
	TIME	0.02	0.22	80.69	146.22
0.7	OBJ	0.03	0.03	0.07	0.03
	TIME	0.02	0.30	110.43	82.51
0.9	OBJ	0.02	0.02	0.06	0.02
	TIME	0.02	0.25	107.96	146.33

Table 12. High-dimensional quantile regression: comparison of the objective values of problem (6) and run time (in seconds) for the microarray data ($n = 120, p = 10,000$) in Scheetz et al. (2006). All the quantities are averaged over 20 independent runs.

τ		HDQR	FHDQR	REHLINE	CVXR
0.1	OBJ	0.02	0.80	0.06	0.03
	TIME	0.05	0.22	1012.93	398.49
0.3	OBJ	0.03	0.03	0.07	0.04
	TIME	0.06	1.80	1274.54	353.56
0.5	OBJ	0.03	0.03	0.07	0.05
	TIME	0.08	1.55	1387.64	431.03
0.7	OBJ	0.03	0.03	0.07	0.04
	TIME	0.07	1.07	1205.02	529.31
0.9	OBJ	0.02	0.02	0.06	0.03
	TIME	0.03	1.13	1195.17	590.80

6. Conclusion and Discussion

In this work, we have developed a finite smoothing algorithm to compute the exact solution of high-dimensional support vector machines and quantile regression. Extensive numerical studies demonstrate our algorithm can be orders of magnitude faster than the existing solvers, even with better precision. We have implemented our algorithms in two R packages, hdsvm and hdqr, available on CRAN.

While our focus has been on unconstrained problems, our FSA framework can readily accommodate linear constraints. This suggests the potential of our algorithm in more complex scenarios such as SVMs with fairness constraints (Zafar et al., 2017) or compositional data analysis (Greenacre, 2021). We leave full investigations to future studies.

Acknowledgements

We thank all the anonymous reviewers, the Area Chair, and the Senior Area Chair for their helpful comments and valuable time reviewing our paper. Their feedback helped us improve the quality of this paper largely.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

Alberto, I. R. I., Alberto, N. R. I., Ghosh, A. K., Jain, B., Jayakumar, S., Martinez-Martin, N., McCague, N., Moukheiber, D., Moukheiber, L., Moukheiber, M., et al. The impact of commercial health datasets on medical research and health-care algorithms. *The Lancet Digital Health*, 5(5):e288–e294, 2023.

Alon, U., Barkai, N., Notterman, D. A., Gish, K., Ybarra, S., Mack, D., and Levine, A. J. Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proceedings of the National Academy of Sciences*, 96(12): 6745–6750, 1999.

Angelopoulos, A. N. and Bates, S. A gentle introduction to conformal prediction and distribution-free uncertainty quantification. *arXiv preprint arXiv:2107.07511*, 2021.

Ardeshir, N., Sanford, C., and Hsu, D. J. Support vector machines and linear regression coincide with very high-dimensional features. *Advances in Neural Information Processing Systems*, 34:4907–4918, 2021.

Bassett, G. W., Koenker, R., Chernozhukov, V., He, X., and Peng, L. A quantile regression memoir. In *Handbook of Quantile Regression*, volume 1. Chapman and Hall/CRC Boca Raton, FL, USA, 2017.

Bertrand, Q. and Massias, M. Anderson acceleration of coordinate descent. In *International Conference on Artificial Intelligence and Statistics*, pp. 1288–1296. PMLR, 2021.

Boyd, S., Parikh, N., Chu, E., Peleato, B., Eckstein, J., et al. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, 3(1):1–122, 2011.

Bradley, P. S. and Mangasarian, O. L. Feature selection via concave minimization and support vector machines. In *ICML*, volume 98, pp. 82–90, 1998.

Chang, K.-W., Hsieh, C.-J., and Lin, C.-J. Coordinate descent method for large-scale l2-loss linear support vector machines. *Journal of Machine Learning Research*, 9(7), 2008.

Chen, C. A finite smoothing algorithm for quantile regression. *Journal of Computational and Graphical Statistics*, 16(1):136–164, 2007.

Chiang, A. P., Beck, J. S., Yen, H.-J., Tayeh, M. K., Scheetz, T. E., Swiderski, R. E., Nishimura, D. Y., Braun, T. A., Kim, K.-Y. A., Huang, J., et al. Homozygosity mapping with snp arrays identifies trim32, an e3 ubiquitin ligase, as a bardet–biedl syndrome gene (bbs11). *Proceedings of the National Academy of Sciences*, 103(16):6287–6292, 2006.

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

Dai, B. and Qiu, Y. ReHLine: Regularized composite ReLU-ReHU loss minimization with linear computation and linear convergence. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

Efron, B., Hastie, T., Johnstone, I., and Tibshirani, R. Least angle regression. *The Annals of Statistics*, 32(2):407–499, 2004.

Fan, J. and Fan, Y. High dimensional classification using features annealed independence rules. *Annals of statistics*, 36(6):2605, 2008.

Fan, J. and Li, R. Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association*, 96(456):1348–1360, 2001.

Friedman, J., Hastie, T., and Tibshirani, R. Regularization paths for generalized linear models via coordinate descent. *Journal of statistical software*, 33(1):1, 2010.

Fu, A., Narasimhan, B., and Boyd, S. CVXR: An R package for disciplined convex optimization. *Journal of Statistical Software*, 94(14):1–34, 2020. doi: 10.18637/jss.v094.i14.

Ghaoui, L. E., Viallon, V., and Rabbani, T. Safe feature elimination for the lasso and sparse supervised learning problems. *arXiv preprint arXiv:1009.4219*, 2010.

Golub, T. R., Slonim, D. K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J. P., Coller, H., Loh, M. L., Downing, J. R., Caligiuri, M. A., et al. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science*, 286(5439):531–537, 1999.

Graham, K., de Las Morenas, A., Tripathi, A., King, C., Kavanah, M., Mendez, J., Stone, M., Slama, J., Miller, M., Antoine, G., et al. Gene expression in histologically normal epithelium from breast cancer patients and from cancer-free prophylactic mastectomy patients shares a similar profile. *British Journal of Cancer*, 102(8):1284–1293, 2010.

Greenacre, M. Compositional data analysis. *Annual Review of Statistics and its Application*, 8:271–299, 2021.

Gu, Y., Fan, J., Kong, L., Ma, S., and Zou, H. ADMM for high-dimensional sparse penalized quantile regression. *Technometrics*, 60(3):319–331, 2018.

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

Hastie, T., Tibshirani, R., Friedman, J. H., and Friedman, J. H. *The elements of statistical learning: data mining, inference, and prediction*, volume 2. Springer, 2009.

He, B. and Yuan, X. On the o(1/n) convergence rate of the douglas–rachford alternating direction method. *SIAM Journal on Numerical Analysis*, 50(2):700–709, 2012.

He, X., Pan, X., Tan, K. M., and Zhou, W.-X. Smoothed quantile regression with large-scale inference. *Journal of Econometrics*, 232(2):367–388, 2023.

Hsieh, C.-J., Chang, K.-W., Lin, C.-J., Keerthi, S. S., and Sundararajan, S. A dual coordinate descent method for large-scale linear svm. In *Proceedings of the 25th international conference on Machine learning*, pp. 408–415, 2008.

Hsu, D., Muthukumar, V., and Xu, J. On the proliferation of support vectors in high dimensions. In *International Conference on Artificial Intelligence and Statistics*, pp. 91–99. PMLR, 2021.

Huang, J., Ma, S., and Zhang, C.-H. Adaptive lasso for sparse high-dimensional regression models. *Statistica Sinica*, pp. 1603–1618, 2008.

Hunter, D. R. and Lange, K. Quantile regression via an MM algorithm. *Journal of Computational and Graphical Statistics*, 9(1):60, 2000. doi: 10.2307/1390613.

Hunter, D. R. and Lange, K. A tutorial on MM algorithms. *The American Statistician*, 58(1):30–37, 2004.

Karimireddy, S. P., Koloskova, A., Stich, S. U., and Jaggi, M. Efficient greedy coordinate descent for composite problems. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 2887–2896. PMLR, 2019.

Koenker, R. *Quantile regression*, volume 38. Cambridge university press, 2005.

Koenker, R. and Bassett, G. W. Regression quantiles. *Econometrica: Journal of the Econometric Society*, pp. 33–50, 1978.

Koenker, R., Portnoy, S., Ng, P. T., Zeileis, A., Grosjean, P., and Ripley, B. D. Package 'quantreg'. *Reference manual available at R-CRAN: https://cran. rproject. org/web/packages/quantreg/quantreg. pdf*, 2018.

Lei, J., G'Sell, M., Rinaldo, A., Tibshirani, R. J., and Wasserman, L. Distribution-free predictive inference for regression. *Journal of the American Statistical Association*, 113(523):1094–1111, 2018.

Luo, Z.-Q. and Tseng, P. On the linear convergence of descent methods for convex essentially smooth minimization. *SIAM Journal on Control and Optimization*, 30(2): 408–425, 1992.

Lv, S., He, X., and Wang, J. A unified penalized method for sparse additive quantile models: an RKHS approach. *Annals of the Institute of Statistical Mathematics*, 69:897–923, 2017.

Mai, Q. and Zou, H. Sparse semiparametric discriminant analysis. *Journal of Multivariate Analysis*, 135:175–188, 2015.

Muthukumar, V., Narang, A., Subramanian, V., Belkin, M., Hsu, D., and Sahai, A. Classification vs regression in overparameterized regimes: Does the loss function matter? *The Journal of Machine Learning Research*, 22(1): 10104–10172, 2021.

Nesterov, Y. Efficiency of coordinate descent methods on huge-scale optimization problems. *SIAM Journal on Optimization*, 22(2):341–362, 2012.

Nutini, J., Laradji, I., and Schmidt, M. Let's make block coordinate descent converge faster: faster greedy rules, message-passing, active-set complexity, and superlinear convergence. *Journal of Machine Learning Research*, 23 (131):1–74, 2022.

Peng, B. and Wang, L. An iterative coordinate descent algorithm for high-dimensional nonconvex penalized quantile regression. *Journal of Computational and Graphical Statistics*, 24(3):676–694, 2015.

Scheetz, T. E., Kim, K.-Y. A., Swiderski, R. E., Philp, A. R., Braun, T. A., Knudtson, K. L., Dorrance, A. M., DiBona, G. F., Huang, J., Casavant, T. L., et al. Regulation of gene expression in the mammalian eye and its relevance to eye disease. *Proceedings of the National Academy of Sciences*, 103(39):14429–14434, 2006.

Shafer, G. and Vovk, V. A tutorial on conformal prediction. *Journal of Machine Learning Research*, 9(3), 2008.

Shalev-Shwartz, S., Singer, Y., and Srebro, N. Pegasos: Primal estimated sub-gradient solver for svm. In *Proceedings of the 24th International Conference on Machine learning*, pp. 807–814, 2007.

Sorace, J. M. and Zhan, M. A data review and re-assessment of ovarian cancer serum proteomic profiling. *BMC bioinformatics*, 4:1–13, 2003.

Tan, K. M., Wang, L., and Zhou, W.-X. High-dimensional quantile regression: Convolution smoothing and concave regularization. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 84(1):205–233, 2022.

Tarzanagh, D. A., Li, Y., Thrampoulidis, C., and Oymak, S. Transformers as support vector machines. *arXiv preprint arXiv:2308.16898*, 2023.

Tibshirani, R. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 58(1):267–288, 1996.

Tibshirani, R., Bien, J., Friedman, J., Hastie, T., Simon, N., Taylor, J., and Tibshirani, R. J. Strong rules for discarding predictors in lasso-type problems. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 74 (2):245–266, 2012.

Tibshirani, R. J., Foygel Barber, R., Candes, E., and Ramdas, A. Conformal prediction under covariate shift. *Advances in Neural Information Processing Systems*, 32, 2019.

Tseng, P. Convergence of a block coordinate descent method for nondifferentiable minimization. *Journal of Optimization Theory and Applications*, 109:475–494, 2001.

Tseng, P. and Yun, S. A coordinate gradient descent method for nonsmooth separable minimization. *Mathematical Programming*, 117:387–423, 2009.

Vapnik, V. *The nature of statistical learning theory*. Springer Science & Business Media, 1999a.

Vapnik, V. N. An overview of statistical learning theory. *IEEE Transactions on Neural Networks*, 10(5):988–999, 1999b.

Wang, B. and Yang, A. A consolidated cross-validation algorithm for support vector machines via data reduction. *Advances in Neural Information Processing Systems*, 35: 394–405, 2022.

Wang, B. and Zou, H. Fast and exact leave-one-out analysis of large-margin classifiers. *Technometrics*, 64(3):291–298, 2022.

Wang, B., Zhou, L., Gu, Y., and Zou, H. Density-convoluted support vector machines for high-dimensional classification. *IEEE Transactions on Information Theory*, 69(4): 2523–2536, 2022.

Wang, L., Zhu, J., and Zou, H. The doubly regularized support vector machine. *Statistica Sinica*, pp. 589–615, 2006.

Wang, L., Zhu, J., and Zou, H. Hybrid huberized support vector machines for microarray classification and gene selection. *Bioinformatics*, 24(3):412–419, 2008.

Wright, S. J. Coordinate descent algorithms. *Mathematical Programming*, 151(1):3–34, 2015.

Wu, T. T. and Lange, K. Coordinate descent algorithms for lasso penalized regression. *The Annals of Applied Statistics*, 2(1):224, 2008.

Yang, Y. and Zou, H. A cocktail algorithm for solving the elastic net penalized Cox's regression in high dimensions. *Statistics and its Interface*, 6(2):167–173, 2013a.

Yang, Y. and Zou, H. An efficient algorithm for computing the hhsvm and its generalizations. *Journal of Computational and Graphical Statistics*, 22(2):396–415, 2013b.

Yi, C. and Huang, J. Semismooth newton coordinate descent algorithm for elastic-net penalized huber loss regression and quantile regression. *Journal of Computational and Graphical Statistics*, 26(3):547–557, 2017.

Zafar, M. B., Valera, I., Rogriguez, M. G., and Gummadi, K. P. Fairness constraints: Mechanisms for fair classification. In *Artificial Intelligence and Statistics*, pp. 962–970. PMLR, 2017.

Zhang, C.-H. Nearly unbiased variable selection under minimax concave penalty. *Annals of Statistics*, 38(2): 894–942, 2010.

Zhang, X., Wu, Y., Wang, L., and Li, R. Variable selection for support vector machines in moderately high dimensions. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 78(1):53–76, 2016.

Zhu, J., Rosset, S., Tibshirani, R., and Hastie, T. 1-norm support vector machines. *Advances in Neural Information Processing Systems*, 16, 2003.

Zou, H. The adaptive lasso and its oracle properties. *Journal of the American statistical association*, 101(476):1418–1429, 2006.

Zou, H. and Hastie, T. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 67(2):301–320, 2005.

Zou, H. and Li, R. One-step sparse estimates in nonconcave penalized likelihood models. *Annals of Statistics*, 36(4): 1509, 2008.

Zou, H. and Zhang, H. H. On the adaptive elastic-net with a diverging number of parameters. *Annals of Statistics*, 37 (4):1733, 2009.

A. Technical Proofs

A.1. Proof of Proposition 2.1

Proof. We have

$$G^\delta(\beta_0, \boldsymbol{\beta}) = \frac{1}{n}\sum_{i=1}^n L_\delta\left(y_i(\beta_0 + \mathbf{x}_i^\top \boldsymbol{\beta})\right) + P_{\boldsymbol{\omega},\lambda_1,\lambda_2}(\boldsymbol{\beta}),$$

where $P_{\boldsymbol{\omega},\lambda_1,\lambda_2}(\boldsymbol{\beta}) \equiv \lambda_1\|\boldsymbol{\omega}\circ\boldsymbol{\beta}\|_1 + \frac{\lambda_2}{2}\|\boldsymbol{\beta}\|_2^2$. By the definition of L_δ, for any $t \in \mathbb{R}$, $0 \le L_\delta(t) - L(t) \le \delta/4$. It follows that

$$0 \le G^\delta(\beta_0, \boldsymbol{\beta}) - G(\beta_0, \boldsymbol{\beta}) \le \delta/4, \;\; \forall \boldsymbol{\beta} \in \mathbb{R}^p, \beta_0 \in \mathbb{R}, \tag{11}$$

Specifically, we have

$$\min_{\beta_0\in\mathbb{R},\boldsymbol{\beta}\in\mathbb{R}^p} G^\delta(\beta_0, \boldsymbol{\beta}) \le G(\widehat{\beta}_0, \widehat{\boldsymbol{\beta}}) + \delta/4.$$

□

A.2. Proof of Theorem 2.3

Proof. The Lagrangian of problem (3) is:

$$L(\boldsymbol{\beta}, \beta_0, \xi_i, \eta_j) = \frac{1}{n}\sum_{i=1}^n L_\delta\left(y_i(\beta_0 + \mathbf{x}_i^\top \boldsymbol{\beta})\right) + P_{\boldsymbol{\omega},\lambda_1,\lambda_2}(\boldsymbol{\beta}) + \sum_{i\in\widehat{S}^\delta} \xi_i(y_i(\beta_0 + \mathbf{x}_i^\top\boldsymbol{\beta}) - 1), \tag{12}$$

where ξ_i's are the Lagrangian multipliers. Note that $(\widehat{\beta}_0^\delta, \widehat{\boldsymbol{\beta}}^\delta)$ is the minimizer of constrained problem (3), thus we have

$$\begin{cases} \frac{1}{n}\sum_i y_i L_\delta'\left(y_i(\widehat{\beta}_0^\delta + \mathbf{x}_i^\top\widehat{\boldsymbol{\beta}}^\delta)\right)\mathbf{x}_i + \lambda_2\widehat{\boldsymbol{\beta}}^\delta + \sum\limits_{i\in\widehat{S}^\delta} y_i\xi_i\mathbf{x}_i + \lambda_1\partial|\widehat{\boldsymbol{\beta}}^\delta| \ni \mathbf{0}, \\ \frac{1}{n}\sum_i y_i L_\delta'\left(y_i(\widehat{\beta}_0^\delta + \mathbf{x}_i^\top\widehat{\boldsymbol{\beta}}^\delta)\right) + \sum\limits_{i\in\widehat{S}^\delta}\xi_i y_i = 0, \\ 1 = y_i(\widehat{\beta}_0^\delta + \mathbf{x}_i^\top\widehat{\boldsymbol{\beta}}^\delta), i \in \widehat{S}^\delta. \end{cases} \tag{13}$$

In particular, there exist a sequence $\{\eta_1, \cdots, \eta_p\}$ such that for the first display of (13), we have

$$\mathbf{0} = \frac{1}{n}\sum_i y_i L_\delta'(y_i(\widehat{\beta}_0^\delta + \mathbf{x}_i^\top\widehat{\boldsymbol{\beta}}^\delta))\mathbf{x}_i + \lambda_2\widehat{\boldsymbol{\beta}}^\delta + \sum_{i\in\widehat{S}^\delta} y_i\xi_i\mathbf{x}_i + \lambda_1\sum_j \eta_j. \tag{14}$$

In this proof, denote $L_h(t) = (1-t)_+$. According to the definition of $L_\delta(t)$ and $L_h(t)$, $\{L_\delta'(y_i(\widehat{\beta}_0^\delta + \mathbf{x}_i^\top\widehat{\boldsymbol{\beta}}^\delta))\} = \partial L_h(y_i(\widehat{\beta}_0^\delta + \mathbf{x}_i^\top\widehat{\boldsymbol{\beta}}^\delta))$ when $i \notin \widehat{S}^\delta$. For $i \in \widehat{S}^\delta$, $\{L_\delta'(y_i(\widehat{\beta}_0^\delta + \mathbf{x}_i^\top\widehat{\boldsymbol{\beta}}^\delta))\} = -\frac{1}{2} \in \partial L_h(y_i(\widehat{\beta}_0^\delta + \mathbf{x}_i^\top\widehat{\boldsymbol{\beta}}^\delta))$. It then follows from (14) that

$$\begin{aligned} \mathbf{0} &= \frac{1}{n}\sum_{i\in\widehat{S}^\delta} y_i L_\delta'(y_i(\widehat{\beta}_0^\delta + \mathbf{x}_i^\top\widehat{\boldsymbol{\beta}}^\delta))\mathbf{x}_i + \frac{1}{n}\sum_{i\notin\widehat{S}^\delta} y_i L_\delta'(y_i(\widehat{\beta}_0^\delta + \mathbf{x}_i^\top\widehat{\boldsymbol{\beta}}^\delta))\mathbf{x}_i + \lambda_2\widehat{\boldsymbol{\beta}}^\delta + \sum_{i\in\widehat{S}^\delta} y_i\xi_i\mathbf{x}_i + \lambda_1\sum_j\eta_j \\ &\in \frac{1}{n}\sum_{i\in\widehat{S}^\delta} y_i\partial L_h(y_i(\widehat{\beta}_0^\delta + \mathbf{x}_i^\top\widehat{\boldsymbol{\beta}}^\delta))\mathbf{x}_i + \frac{1}{n}\sum_{i\notin\widehat{S}^\delta} y_i\partial L_h(y_i(\widehat{\beta}_0^\delta + \mathbf{x}_i^\top\widehat{\boldsymbol{\beta}}^\delta))\mathbf{x}_i + \lambda_2\widehat{\boldsymbol{\beta}}^\delta + \sum_{i\in\widehat{S}^\delta} y_i\xi_i\mathbf{x}_i + \lambda_1\sum_j\eta_j \\ &= \frac{1}{n}\sum_i y_i\partial L_h(y_i(\widehat{\beta}_0^\delta + \mathbf{x}_i^\top\widehat{\boldsymbol{\beta}}^\delta))\mathbf{x}_i + \lambda_2\widehat{\boldsymbol{\beta}}^\delta + \sum_{i\in\widehat{S}^\delta} y_i\xi_i\mathbf{x}_i + \lambda_1\sum_j\eta_j \end{aligned}$$

Similarly, we have $0 \in \frac{1}{n}\sum_i y_i\partial L_h(y_i(\widehat{\beta}_0^\delta + \mathbf{x}_i^\top\widehat{\boldsymbol{\beta}}^\delta)) + \sum\limits_{i\in\widehat{S}^\delta}\xi_i y_i$, indicating that $(\widehat{\beta}_0^\delta, \widehat{\boldsymbol{\beta}}^\delta)$ satisfies the KKT condition of the following constrained problem

$$\begin{gathered} \min \;\frac{1}{n}\sum_{i=1}^n L_h\left(y_i(\beta_0 + \mathbf{x}_i^\top\boldsymbol{\beta})\right) + P_{\boldsymbol{\omega},\lambda_1,\lambda_2}(\boldsymbol{\beta}) \\ \text{subject to } \; 1 = y_i(\beta_0 + \mathbf{x}_i^\top\boldsymbol{\beta}),\; i \in \widehat{S}^\delta, \end{gathered} \tag{15}$$

thus $(\widehat{\beta}_0^\delta, \widehat{\boldsymbol{\beta}}^\delta)$ is the minimizer of problem (15). Moreover, it can be easily seen that $(\widehat{\beta}_0^\delta, \widehat{\boldsymbol{\beta}}^\delta)$ is a feasible point of problem (15), this implies that

$$G(\widehat{\beta}_0^\delta, \widehat{\boldsymbol{\beta}}^\delta) \leqslant G(\widehat{\beta}_0, \widehat{\boldsymbol{\beta}}) \leqslant G(\widehat{\beta}_0^\delta, \widehat{\boldsymbol{\beta}}^\delta).$$

Thus we have $(\widehat{\beta}_0, \widehat{\boldsymbol{\beta}}) = (\widehat{\beta}_0^\delta, \widehat{\boldsymbol{\beta}}^\delta)$. □

A.3. Proof of Theorem 2.4

Proof. From Proposition 2.1, we know,

$$0 \leqslant G^\delta(\beta_0, \boldsymbol{\beta}) - G(\beta_0, \boldsymbol{\beta}) < \frac{\delta}{4}, \forall \boldsymbol{\beta} \in \mathbb{R}^p, \beta_0 \in \mathbb{R}. \tag{16}$$

Thus $\delta < \delta^\sharp < 4\eta$ gives $0 \leqslant G^\delta(\beta_0, \boldsymbol{\beta}) - G(\beta_0, \boldsymbol{\beta}) < \eta$. Since $(\widehat{\beta}_0, \widehat{\boldsymbol{\beta}})$ is a feasible point of problem (4), using the optimality of $(\widetilde{\beta}_0^\delta, \widetilde{\boldsymbol{\beta}}^\delta)$, we have $G^\delta(\widetilde{\beta}_0^\delta, \widetilde{\boldsymbol{\beta}}^\delta) \le G^\delta(\widehat{\beta}_0, \widehat{\boldsymbol{\beta}})$. It then follows that

$$G(\widetilde{\beta}_0^\delta, \widetilde{\boldsymbol{\beta}}^\delta) - G(\widehat{\beta}_0, \widehat{\boldsymbol{\beta}}) = [G(\widetilde{\beta}_0^\delta, \widetilde{\boldsymbol{\beta}}^\delta) - G^\delta(\widetilde{\beta}_0^\delta, \widetilde{\boldsymbol{\beta}}^\delta)] + [G^\delta(\widetilde{\beta}_0^\delta, \widetilde{\boldsymbol{\beta}}^\delta) - G^\delta(\widehat{\beta}_0, \widehat{\boldsymbol{\beta}})] + [G^\delta(\widehat{\beta}_0, \widehat{\boldsymbol{\beta}}) - G(\widehat{\beta}_0, \widehat{\boldsymbol{\beta}})] < \eta,$$

indicating that $(\widetilde{\beta}_0^\delta, \widetilde{\boldsymbol{\beta}}^\delta) \notin C_{\delta_0/2}$ by the definition of $C_{\delta_0/2}$, therefore $\left|\widetilde{\beta}_0^\delta + \mathbf{x}_i^\top \widetilde{\boldsymbol{\beta}}^\delta - \widehat{\beta}_0 - \mathbf{x}_i^\top \widehat{\boldsymbol{\beta}}\right| < \delta_0/2$ for all i. Furthermore, for any $i \in \widetilde{S}^\delta$,

$$\begin{aligned}
\left|1 - y_i(\mathbf{x}_i^\top \widehat{\boldsymbol{\beta}} + \widehat{\beta}_0)\right| &\leqslant \left|1 - y_i(\mathbf{x}_i^\top \widetilde{\boldsymbol{\beta}}^\delta + \widetilde{\beta}_0^\delta)\right| + \left|y_i(\mathbf{x}_i^\top \widetilde{\boldsymbol{\beta}}^\delta + \widetilde{\beta}_0^\delta) - y_i(\mathbf{x}_i^\top \widehat{\boldsymbol{\beta}} + \widehat{\beta}_0)\right| \\
&= \left|1 - y_i(\mathbf{x}_i^\top \widetilde{\boldsymbol{\beta}}^\delta + \widetilde{\beta}_0^\delta)\right| + |y_i| \left|(\mathbf{x}_i^\top \widetilde{\boldsymbol{\beta}}^\delta + \widetilde{\beta}_0^\delta) - (\mathbf{x}_i^\top \widehat{\boldsymbol{\beta}} + \widehat{\beta}_0)\right| \\
&< \delta + \delta_0/2 < \delta_0,
\end{aligned}$$

which implies that $i \in S^\star$. We conclude that $\widetilde{S} \subseteq \widetilde{S}^\delta \subseteq S^\star$. □

B. Iteration Complexity Analysis of the GCD Algorithm or High-Dimensional SVM

Note that the intercept term $\boldsymbol{\beta}_0$ can be absorbed into the formulation by setting $x_{i1} = 1$ for $i = 1, \cdots, n$ and $w_1 = 0$. We let $G(\boldsymbol{\beta}) = \frac{1}{n}\sum_{i=1}^n L_\delta\left(y_i \mathbf{x}_i^\top \boldsymbol{\beta}\right) + P_{\boldsymbol{\omega}, \lambda_1, \lambda_2}(\boldsymbol{\beta})$ rewrite problem (3) as the following constrained convex optimization problem.

$$\min_{\boldsymbol{\beta} \in \mathbb{R}^{p+1}} G(\boldsymbol{\beta}), \text{ subject to } \mathbf{1}_{|\widetilde{S}|} = \boldsymbol{y}_s \mathbf{x}_s^\top \boldsymbol{\beta}, \tag{17}$$

where $\boldsymbol{y}_s := \left\{\boldsymbol{y}_i; i \in \widetilde{S}\right\}$ and $\mathbf{x}_s := \left\{\mathbf{x}_i; i \in \widetilde{S}\right\}$. Denote $\boldsymbol{\beta}^*$ is the optimal solution of (17).

The augmented Lagrangian function of problem (17) is

$$L_\sigma(\boldsymbol{\beta}, \boldsymbol{\theta}) = G(\boldsymbol{\beta}) + < \boldsymbol{\theta}, \mathbf{1} - \boldsymbol{y}_s \mathbf{x}_s^\top \boldsymbol{\beta} > + \frac{\sigma}{2}\left\|\boldsymbol{y}_s \mathbf{x}_s^\top \boldsymbol{\beta} - \mathbf{1}\right\|_2^2.$$

Let $\boldsymbol{r}^k = \boldsymbol{y}_s \mathbf{x}_s^\top \boldsymbol{\beta}^k - \mathbf{1}$. By definition, $\boldsymbol{\beta}^{k+1}$ minimizes $L_\sigma\left(\boldsymbol{\beta}, \boldsymbol{\theta}^k\right)$ that implies

$$\mathbf{0} \in \partial G(\boldsymbol{\beta}) - \boldsymbol{y}_s \mathbf{x}_s^\top \left(\boldsymbol{\theta}^k - \sigma \boldsymbol{r}^{k+1}\right).$$

Since $\boldsymbol{\theta}^{k+1} = \boldsymbol{\theta}^k - \sigma \boldsymbol{r}^{k+1}$, we can obtain that $\mathbf{0} \in \partial G(\boldsymbol{\beta}) - \boldsymbol{y}_s \mathbf{x}_s^\top \boldsymbol{\theta}^{k+1}$. That implies that $\boldsymbol{\beta}^{k+1}$ minimizes $G(\boldsymbol{\beta}) - \left(\boldsymbol{\theta}^{k+1}\right)^\top \boldsymbol{y}_s \mathbf{x}_s \beta$. Then it follows that

$$G(\boldsymbol{\beta}^{k+1}) - \left(\boldsymbol{\theta}^{k+1}\right)^\top \boldsymbol{y}_s \mathbf{x}_s \boldsymbol{\beta}^{k+1} \leqslant G(\boldsymbol{\beta}^*) - \left(\boldsymbol{\theta}^{k+1}\right)^\top \boldsymbol{y}_s \mathbf{x}_s \boldsymbol{\beta}^*. \tag{18}$$

By the optimality of $\boldsymbol{\beta}^*$ in problem (17), we have $\mathbf{0} \in \partial G(\boldsymbol{\beta}^*) - \boldsymbol{y}_s \mathbf{x}_s^\top \boldsymbol{\theta}^*$ and $\mathbf{1} = y_i \mathbf{x}_i^\top \boldsymbol{\beta}, \forall i \in S$. It means that $\boldsymbol{\beta}^*$ minimizes $G(\boldsymbol{\beta}) - \boldsymbol{y}_s \mathbf{x}_s^\top \boldsymbol{\theta}^*$.

Then we have

$$G(\boldsymbol{\beta}^{k+1}) - (\boldsymbol{\theta}^*)^\top \boldsymbol{y}_s \mathbf{x}_s \boldsymbol{\beta}^{k+1} \geqslant G(\boldsymbol{\beta}^*) - (\boldsymbol{\theta}^*)^\top \boldsymbol{y}_s \mathbf{x}_s \boldsymbol{\beta}^*. \tag{19}$$

Combining equations (18) and (19), and after multiplying the combined outcome by 2, results in,

$$2\left(\boldsymbol{\theta}^{k+1} - \boldsymbol{\theta}^*\right)^\top \boldsymbol{y}_s \mathbf{x}_s \left(\boldsymbol{\beta}^{k+1} - \boldsymbol{\beta}^*\right) \geqslant 0.$$

Appling $\boldsymbol{\theta}^{k+1} = \boldsymbol{\theta}^k - \sigma \boldsymbol{r}^{k+1}$ and $\boldsymbol{r}^{k+1} = \boldsymbol{y}_s \mathbf{x}_s \boldsymbol{\beta}^{k+1} - \mathbf{1} = \boldsymbol{y}_s \mathbf{x}_s \boldsymbol{\beta}^{k+1} - \boldsymbol{y}_s \mathbf{x}_s \boldsymbol{\beta}^*$, we have

$$2\left(\boldsymbol{\theta}^* - \boldsymbol{\theta}^k\right)^\top \boldsymbol{r}^{k+1} + \sigma \left\|\boldsymbol{r}^{k+1}\right\|_2^2 + \sigma \left\|\boldsymbol{r}^{k+1}\right\|_2^2 \leq 0.$$

Given $\boldsymbol{r}^{k+1} = \frac{1}{\sigma}\left(\boldsymbol{\theta}^k - \boldsymbol{\theta}^{k+1}\right)$, it can be rewritten as

$$\frac{2}{\sigma}\left(\boldsymbol{\theta}^k - \boldsymbol{\theta}^*\right)^\top \left(\boldsymbol{\theta}^{k+1} - \boldsymbol{\theta}^k\right) + \frac{1}{\sigma}\left\|\boldsymbol{\theta}^{k+1} - \boldsymbol{\theta}^k\right\|_2^2 + \sigma \left\|\boldsymbol{r}^{k+1}\right\|_2^2 \leq 0.$$

Furthermore, $\boldsymbol{\theta}^{k+1} - \boldsymbol{\theta}^k = \left(\boldsymbol{\theta}^{k+1} - \boldsymbol{\theta}^*\right) - \left(\boldsymbol{\theta}^k - \boldsymbol{\theta}^*\right)$ gives us

$$\frac{1}{\sigma}\left\|\boldsymbol{\theta}^{k+1} - \boldsymbol{\theta}^*\right\|_2^2 \leq \frac{1}{\sigma}\left\|\boldsymbol{\theta}^k - \boldsymbol{\theta}^*\right\|_2^2 - \sigma \left\|\boldsymbol{r}^{k+1}\right\|_2^2. \tag{20}$$

This shows that $1/\sigma \left\|\boldsymbol{\theta}^k - \boldsymbol{\theta}^*\right\|_2^2$ is a non-increasing sequence.

Because $1/\sigma \left\|\boldsymbol{\theta}^k - \boldsymbol{\theta}^*\right\|_2^2 \leq 1/\sigma \left\|\boldsymbol{\theta}^0 - \boldsymbol{\theta}^*\right\|_2^2$, it follows that $\boldsymbol{\theta}^k$ are bounded. Iterating the above inequality gives that

$$\sigma \sum_{k=0}^{\infty} \left\|\boldsymbol{r}^{k+1}\right\|_2^2 \leq 1/\sigma \left\|\boldsymbol{\theta}^0 - \boldsymbol{\theta}^*\right\|_2^2,$$

which implies that $\boldsymbol{r}^k \to 0$ as $k \to \infty$.

Meanwhile the inequality (18) can be rewritten as

$$G(\boldsymbol{\beta}^{k+1}) - G(\boldsymbol{\beta}^*) \leqslant \left(\boldsymbol{\theta}^{k+1}\right)^\top \boldsymbol{y}_s \mathbf{x}_s \boldsymbol{\beta}^{k+1} - \left(\boldsymbol{\theta}^{k+1}\right)^\top \boldsymbol{y}_s \mathbf{x}_s \boldsymbol{\beta}^* = \left(\boldsymbol{\theta}^{k+1}\right)^\top \boldsymbol{r}^{k+1}. \tag{21}$$

Since $\boldsymbol{\theta}^k$ is bounded and $\boldsymbol{r}^{k+1}$ goes to zero, the right side in (21) goes to zero. Similarly, the inequality (19) can be rewritten as

$$G(\boldsymbol{\beta}^{k+1}) - G(\boldsymbol{\beta}^*) \geqslant (\boldsymbol{\theta}^*)^\top \boldsymbol{y}_s \mathbf{x}_s \boldsymbol{\beta}^{k+1} - (\boldsymbol{\theta}^*)^\top \boldsymbol{y}_s \mathbf{x}_s \boldsymbol{\beta}^* = (\boldsymbol{\theta}^*)^\top \boldsymbol{r}^{k+1}, \tag{22}$$

the right side also goes to zero as $k \to \infty$. Therefore $\lim_{k\to\infty} G(\boldsymbol{\beta}^{k+1}) - G(\boldsymbol{\beta}^*) = 0$. By definition, it means objective convergence.

Applying $\boldsymbol{r}^{k+1} = 1/\sigma\left(\boldsymbol{\theta}^k - \boldsymbol{\theta}^{k+1}\right)$, we rewrite the inequality (20) as

$$\left\|\boldsymbol{\theta}^{k+1} - \boldsymbol{\theta}^*\right\|_2^2 \leq \left\|\boldsymbol{\theta}^k - \boldsymbol{\theta}^*\right\|_2^2 - \left\|\boldsymbol{\theta}^k - \boldsymbol{\theta}^{k+1}\right\|_2^2. \tag{23}$$

It follows that

$$\sum_{k=0}^{\infty}\left\|\boldsymbol{\theta}^{k}-\boldsymbol{\theta}^{k+1}\right\|_2^2 \leq \left\|\boldsymbol{\theta}^{0}-\boldsymbol{\theta}^{*}\right\|_2^2 . \tag{24}$$

Recall that we proved $\boldsymbol{\beta}^{k+1}$ minimizes $G(\boldsymbol{\beta})-\left(\boldsymbol{\theta}^{k+1}\right)^{\top}\boldsymbol{y}_s\mathbf{x}_s\boldsymbol{\beta}$, then we obtain

$$G\left(\boldsymbol{\beta}^{k+1}\right)-\left(\boldsymbol{\theta}^{k+1}\right)^{\top}\boldsymbol{y}_s\mathbf{x}_s\boldsymbol{\beta}^{k+1} \leqslant G\left(\boldsymbol{\beta}^{k+2}\right)-\left(\boldsymbol{\theta}^{k+1}\right)^{\top}\boldsymbol{y}_s\mathbf{x}_s\boldsymbol{\beta}^{k+2}. \tag{25}$$

Similarly we have

$$G\left(\boldsymbol{\beta}^{k+2}\right)-\left(\boldsymbol{\theta}^{k+2}\right)^{\top}\boldsymbol{y}_s\mathbf{x}_s\boldsymbol{\beta}^{k+2} \leq G\left(\boldsymbol{\beta}^{k+1}\right)-\left(\boldsymbol{\theta}^{k+2}\right)^{\top}\boldsymbol{y}_s\mathbf{x}_s\boldsymbol{\beta}^{k+1}. \tag{26}$$

Adding up (25) and (26) gives

$$\left(\boldsymbol{\theta}^{k+1}-\boldsymbol{\theta}^{k+2}\right)^{\top}\boldsymbol{y}_s\left(\mathbf{x}_s\boldsymbol{\beta}^{k+1}-\mathbf{x}_s\boldsymbol{\beta}^{k+2}\right) \geqslant 0.$$

Applying $\boldsymbol{r}^{k+1}=\boldsymbol{y}_s\mathbf{x}_s\boldsymbol{\beta}^{k+1}-\mathbf{1}$, we have

$$\left(\boldsymbol{\theta}^{k+1}-\boldsymbol{\theta}^{k+2}\right)^{\top}\left(\boldsymbol{r}^{k+1}-\boldsymbol{r}^{k+2}\right) \geqslant 0.$$

Then $\boldsymbol{r}^{k+1}=\frac{1}{\sigma}\left(\boldsymbol{\theta}^{k}-\boldsymbol{\theta}^{k+1}\right)$ gives

$$\frac{1}{\sigma}\left(\boldsymbol{\theta}^{k+1}-\boldsymbol{\theta}^{k+2}\right)^{\top}\left[\left(\boldsymbol{\theta}^{k}-\boldsymbol{\theta}^{k+1}\right)-\left(\boldsymbol{\theta}^{k+1}-\boldsymbol{\theta}^{k+2}\right)\right] \geqslant 0.$$

Since $\|\boldsymbol{a}\|_2^2-\|\boldsymbol{b}\|_2^2=2\boldsymbol{a}^{\top}(\boldsymbol{a}-\boldsymbol{b})-\|\boldsymbol{a}-\boldsymbol{b}\|_2^2$ holds for any two vectors $\boldsymbol{a}$ and $\boldsymbol{b}$, where $\boldsymbol{a}$ and $\boldsymbol{b}$ have the same dimension. Setting $\boldsymbol{a}=\boldsymbol{\theta}^{k}-\boldsymbol{\theta}^{k+1}$ and $\boldsymbol{b}=\boldsymbol{\theta}^{k+1}-\boldsymbol{\theta}^{k+2}$, we howe

$$\begin{aligned}
&\frac{1}{\sigma}\left\|\boldsymbol{\theta}^{k}-\boldsymbol{\theta}^{k+1}\right\|_2^2-\frac{1}{\sigma}\left\|\boldsymbol{\theta}^{k+1}-\boldsymbol{\theta}^{k+2}\right\|_2^2\\
=&\frac{2}{\sigma}\left(\boldsymbol{\theta}^{k}-\boldsymbol{\theta}^{k+1}\right)^{\top}\left[\left(\boldsymbol{\theta}^{k}-\boldsymbol{\theta}^{k+1}\right)-\left(\boldsymbol{\theta}^{k+1}-\boldsymbol{\theta}^{k+2}\right)\right]-\frac{1}{\sigma}\left\|\left(\boldsymbol{\theta}^{k}-\boldsymbol{\theta}^{k+1}\right)-\left(\boldsymbol{\theta}^{k+1}-\boldsymbol{\theta}^{k+2}\right)\right\|_2^2\\
\geqslant&\frac{2}{\sigma}\left(\boldsymbol{\theta}^{k}-\boldsymbol{\theta}^{k+1}\right)^{\top}\left[\left(\boldsymbol{\theta}^{k}-\boldsymbol{\theta}^{k+1}\right)-\left(\boldsymbol{\theta}^{k+1}-\boldsymbol{\theta}^{k+2}\right)\right]\\
&-\frac{2}{\sigma}\left(\boldsymbol{\theta}^{k+1}-\boldsymbol{\theta}^{k+2}\right)^{\top}\left[\left(\boldsymbol{\theta}^{k}-\boldsymbol{\theta}^{k+1}\right)-\left(\boldsymbol{\theta}^{k+1}-\boldsymbol{\theta}^{k+2}\right)\right]\\
&-\frac{1}{\sigma}\left\|\left(\boldsymbol{\theta}^{k}-\boldsymbol{\theta}^{k+1}\right)-\left(\boldsymbol{\theta}^{k+1}-\boldsymbol{\theta}^{k+2}\right)\right\|_2^2\\
=&\frac{2}{\sigma}\left\|\left(\boldsymbol{\theta}^{k}-\boldsymbol{\theta}^{k+1}\right)-\left(\boldsymbol{\theta}^{k+1}-\boldsymbol{\theta}^{k+2}\right)\right\|_2^2-\frac{1}{\sigma}\left\|\left(\boldsymbol{\theta}^{k}-\boldsymbol{\theta}^{k+1}\right)-\left(\boldsymbol{\theta}^{k+1}-\boldsymbol{\theta}^{k+2}\right)\right\|_2^2\\
=&\frac{1}{\sigma}\left\|\left(\boldsymbol{\theta}^{k}-\boldsymbol{\theta}^{k+1}\right)-\left(\boldsymbol{\theta}^{k+1}-\boldsymbol{\theta}^{k+2}\right)\right\|_2^2 \geqslant 0.
\end{aligned}$$

It implies that $\left\{\left\|\boldsymbol{\theta}^{k}-\boldsymbol{\theta}^{k+1}\right\|_2^2\right\}$ is monotonically non -increasing.

Furthermore, we have

$$(k+1)\left\|\boldsymbol{\theta}^{k}-\boldsymbol{\theta}^{k+1}\right\|_{2}^{2} \leq \sum_{t=0}^{k}\left\|\boldsymbol{\theta}^{t}-\boldsymbol{\theta}^{t+1}\right\|_{2}^{2} \tag{27}$$

Therefore, applying inequality (24) and (27) gives

$$\left\|\boldsymbol{\theta}^{k}-\boldsymbol{\theta}^{k+1}\right\|_{2}^{2} \leq \frac{1}{k+1}\left\|\boldsymbol{\theta}^{0}-\boldsymbol{\theta}^{*}\right\|_{2}^{2}.$$

then we have $\left\|\boldsymbol{\theta}^{k}-\boldsymbol{\theta}^{k+1}\right\|_{2}^{2}=O(1/k)$ as $k\to\infty$.

C. Algorithm for Quantile Regression

In this section, we demonstrate how the smoothed quantile regression model is capable of yielding the exact solution to the original quantile regression model.

Define $Q^{\delta}(\beta_0,\boldsymbol{\beta})=\frac{1}{n}\sum_{i=1}^{n}H_{\delta,\tau}(y_i-\beta_0-\mathbf{x}_i^{\top}\boldsymbol{\beta})+\lambda_1\|\boldsymbol{\omega}\circ\boldsymbol{\beta}\|_1+\frac{\lambda_2}{2}\|\boldsymbol{\beta}\|_2^2$. Lemma C.1 tells us if $E^{\star}$ were known, the exact solution could be attained by solving a constrained convex optimization problem, where

$$E^{\star}=\left\{i:\left|y_i-(\widehat{\beta}_0^{qr}+\mathbf{x}_i^{\top}\widehat{\boldsymbol{\beta}}^{qr})\right|=0\right\},$$

and $(\widehat{\beta}_0^{qr},\widehat{\boldsymbol{\beta}}^{qr})$ is the exact quantile regression solution in problem (6).

Lemma C.1. *If $E^{\star}$ is known, define*

$$\begin{aligned}(\widehat{\beta}_0^{\delta},\widehat{\boldsymbol{\beta}}^{\delta})=&\underset{\beta_0\in\mathbb{R},\boldsymbol{\beta}\in\mathbb{R}^p}{\operatorname{argmin}}\;Q^{\delta}(\beta_0,\boldsymbol{\beta}),\\ \textit{subject to}\;\;& y_i=\beta_0+\mathbf{x}_i^{\top}\boldsymbol{\beta},\; i\in E^{\star},\end{aligned}$$

then $(\widehat{\beta}_0^{qr},\widehat{\boldsymbol{\beta}}^{qr})=(\widehat{\beta}_0^{\delta},\widehat{\boldsymbol{\beta}}^{\delta})$ holds.

Proof. The proof of Lemma C.1 bears similarity to that of Theorem C.2, and therefore, for brevity, it is not reiterated here. □

In practice, $E^{\star}$ remains unknown, prompting us to propose a relaxed version of Lemma C.1. This adaptation demonstrates that a subset of $E^{\star}$ is sufficient to accurately derive the exact solution of the quantile regression problem. Specifically, let $\gamma_0=\min_{i\notin E^{\star}}\{|y_i-(\widehat{\beta}_0^{qr}+\mathbf{x}_i^{\top}\widehat{\boldsymbol{\beta}}^{qr})|\}>0$, $C_{\gamma_0/2}=\{(\beta_0,\boldsymbol{\beta}):\|\beta_0\mathbf{1}_n+\mathbf{x}^{\top}\boldsymbol{\beta}-\widehat{\beta}_0^{qr}\mathbf{1}_n-\mathbf{x}^{\top}\widehat{\boldsymbol{\beta}}^{qr}\|_{\infty}\geqslant\gamma_0/2\}$, $\rho=\inf_{(\beta_0,\boldsymbol{\beta})\in C_{\gamma_0/2}}\{Q(\beta_0,\boldsymbol{\beta})-Q(\widehat{\beta}_0^{qr},\widehat{\boldsymbol{\beta}}^{qr})\}$, and $\delta^{*}=\min\{\gamma_0/2,4\rho\}$ and present the following theorem.

Theorem C.2. *For any $\delta\in(0,\delta^{*})$, we can find a set $\widehat{E}^{\delta}\subseteq E^{\star}$ such that $\widetilde{E}^{\delta}=\widehat{E}^{\delta}$, where*

$$\begin{aligned}(\widehat{\beta}_0^{\delta},\widehat{\boldsymbol{\beta}}^{\delta})=&\underset{\beta_0\in\mathbb{R},\boldsymbol{\beta}\in\mathbb{R}^p}{\operatorname{argmin}}\;Q^{\delta}(\beta_0,\boldsymbol{\beta}),\\ \textit{subject to}\;\;& y_i=\beta_0+\mathbf{x}_i^{\top}\boldsymbol{\beta},\; i\in \widehat{E}^{\delta},\end{aligned}\tag{28}$$

and $\widetilde{E}^{\delta}=\{i:-\delta\leq y_i-(\widehat{\beta}_0^{\delta}+\mathbf{x}_i^{\top}\widehat{\boldsymbol{\beta}}^{\delta})\leq\delta\}$. then $(\widehat{\beta}_0^{qr},\widehat{\boldsymbol{\beta}}^{qr})=(\widehat{\beta}_0^{\delta},\widehat{\boldsymbol{\beta}}^{\delta})$ holds for any $\delta\in(0,\delta^{*})$.

Proof. By the definition of $H_{\delta,\tau}$, for any $t\in\mathbb{R}$, $0\leq H_{\delta,\tau}(t)-\rho_\tau(t)\leq\delta/4$. It follows that

$$0\leq Q^{\delta}(\beta_0,\boldsymbol{\beta})-Q(\beta_0,\boldsymbol{\beta})\leq\delta/4,\;\;\forall\boldsymbol{\beta}\in\mathbb{R}^p,\beta_0\in\mathbb{R}. \tag{29}$$

Specifically, we have

$$\min_{\beta_0\in\mathbb{R},\boldsymbol{\beta}\in\mathbb{R}^p}Q^{\delta}(\beta_0,\boldsymbol{\beta})\leq Q(\widehat{\beta}_0^{qr},\widehat{\boldsymbol{\beta}}^{qr})+\delta/4.$$

The Lagrangian of problem (28) is:

$$L(\boldsymbol{\beta}, \beta_0, \xi_i, \eta_j) = \frac{1}{n}\sum_{i=1}^{n} H_{\delta,\tau}\left(y_i - \beta_0 - \mathbf{x}_i^\top \boldsymbol{\beta}\right) + P_{\boldsymbol{\omega},\lambda_1,\lambda_2}(\boldsymbol{\beta}) + \sum_{i\in\widehat{E}^\delta} \xi_i(\beta_0 + \mathbf{x}_i^\top\boldsymbol{\beta} - y_i),$$

where ξ_i's are the Lagrangian multipliers. Since $(\widehat{\beta}_0^\delta, \widehat{\boldsymbol{\beta}}^\delta)$ is the optimal solution of problem (28), we have

$$\begin{cases} -\frac{1}{n}\sum_i H'_{\delta,\tau}\left(y_i - \widehat{\beta}_0^\delta \ - \mathbf{x}_i^\top \widehat{\boldsymbol{\beta}}^\delta\right)\mathbf{x}_i + \lambda_1\partial|\widehat{\boldsymbol{\beta}}^\delta| + \lambda_2\widehat{\boldsymbol{\beta}}^\delta + \sum\limits_{i\in\widehat{E}^\delta}\xi_i\mathbf{x}_i \ni \mathbf{0}, \\ -\frac{1}{n}\sum_i H'_{\delta,\tau}\left(y_i - \widehat{\beta}_0^\delta \ - \mathbf{x}_i^\top \widehat{\boldsymbol{\beta}}^\delta\right) + \sum\limits_{i\in\widehat{E}^\delta}\xi_i = 0 \\ y_i = \widehat{\beta}_0^\delta + \mathbf{x}_i^\top\widehat{\boldsymbol{\beta}}^\delta, i \in \widehat{E}^\delta. \end{cases} \tag{30}$$

In particular, there exist a sequence $\{\eta_1, \cdots, \eta_p\}$ such that

$$-\frac{1}{n}\sum_i H'_{\delta,\tau}\left(y_i - \widehat{\beta}_0^\delta \ - \mathbf{x}_i^\top \widehat{\boldsymbol{\beta}}^\delta\right)\mathbf{x}_i + \lambda_1\sum_j \eta_j + \lambda_2\widehat{\boldsymbol{\beta}}^\delta + \sum_{i\in\widehat{E}^\delta}\xi_i\mathbf{x}_i = \mathbf{0} \tag{31}$$

By the definition of $\rho_\tau(t)$ and $H_{\delta,\tau}(t)$, $\{H'_{\delta,\tau}(y_i - \widehat{\beta}_0^\delta - \mathbf{x}_i^\top\widehat{\boldsymbol{\beta}}^\delta)\} = \partial\rho_\tau(y_i - \widehat{\beta}_0^\delta - \mathbf{x}_i^\top\widehat{\boldsymbol{\beta}}^\delta)$ when $i \notin \widehat{E}^\delta$ and $\{H'_{\delta,\tau}(y_i - \widehat{\beta}_0^\delta - \mathbf{x}_i^\top\widehat{\boldsymbol{\beta}}^\delta)\} = \tau - \frac{1}{2} \in \partial\rho_\tau(y_i - \widehat{\beta}_0^\delta + \mathbf{x}_i^\top\widehat{\boldsymbol{\beta}}^\delta)$ when $i \in \widehat{E}^\delta$. Therefore, it follows from (31)

$$\mathbf{0} \in -\frac{1}{n}\sum_i \partial\rho_\tau\left(y_i - \widehat{\beta}_0^\delta \ - \mathbf{x}_i^\top \widehat{\boldsymbol{\beta}}^\delta\right)\mathbf{x}_i + \lambda_1\sum_j \eta_j + \lambda_2\widehat{\boldsymbol{\beta}}^\delta + \sum_{i\in\widehat{E}^\delta}\xi_i\mathbf{x}_i$$

Similarly, we know $0 \in -\frac{1}{n}\sum_i \partial\rho_\tau\left(y_i - \widehat{\beta}_0^\delta \ - \mathbf{x}_i^\top \widehat{\boldsymbol{\beta}}^\delta\right) + \sum\limits_{i\in\widehat{E}^\delta}\xi_i$. It follows that $(\widehat{\beta}_0^\delta, \widehat{\boldsymbol{\beta}}^\delta)$ satisfies the KKT condition of the constrained problem

$$\begin{aligned} \min \ \ & \frac{1}{n}\sum_{i=1}^{n}\rho_\tau\left(y_i - \beta_0 - \mathbf{x}_i^\top\boldsymbol{\beta}\right) + P_{\boldsymbol{\omega},\lambda_1,\lambda_2}(\boldsymbol{\beta}) \\ & \text{subject to } \ y_i = \beta_0 + \mathbf{x}_i^\top\boldsymbol{\beta},\ i \in \widehat{E}^\delta, \end{aligned} \tag{32}$$

thus it is the minimizer of problem (32). Moreover, $(\widehat{\beta}_0^\delta, \widehat{\boldsymbol{\beta}}^\delta)$ satisfies these equality constraints by $\widehat{E}^\delta \in E^\star$, this implies that $Q(\widehat{\beta}_0^\delta, \widehat{\boldsymbol{\beta}}^\delta) \leqslant Q(\widehat{\beta}_0^{qr}, \widehat{\boldsymbol{\beta}}^{qr}) \leqslant G(\widehat{\beta}_0^\delta, \widehat{\boldsymbol{\beta}}^\delta)$. Thus we have $(\widehat{\beta}_0^{qr}, \widehat{\boldsymbol{\beta}}^{qr}) = (\widehat{\beta}_0^\delta, \widehat{\boldsymbol{\beta}}^\delta)$. □

Through the iterative application of Theorem C.3, the set $\widehat{E}^\delta$ is progressively realized within a finite number of steps.

Theorem C.3. *For any set $\widetilde{E} \subseteq E^\star$ and any $\delta \in (0, \delta^*)$, define*

$$\begin{aligned} (\widetilde{\beta}_0^\delta, \widetilde{\boldsymbol{\beta}}^\delta) = & \operatorname*{argmin}_{\beta_0\in\mathbb{R},\boldsymbol{\beta}\in\mathbb{R}^p} Q^\delta(\beta_0, \boldsymbol{\beta}), \\ & \textit{subject to } \ y_i = \beta_0 + \mathbf{x}_i^\top\boldsymbol{\beta},\ i \in \widetilde{E}, \end{aligned} \tag{33}$$

and let $\widetilde{E}^\delta = \{i\colon -\delta \le y_i - (\widetilde{\beta}_0^\delta + \mathbf{x}_i^\top\widetilde{\boldsymbol{\beta}}^\delta) \le \delta\}$, then the following holds: $\widetilde{E} \subseteq \widetilde{E}^\delta \subseteq E^\star$.

Proof. Applying (29), we have

$$0 \leqslant Q^\delta(\beta_0, \boldsymbol{\beta}) - Q(\beta_0, \boldsymbol{\beta}) < \frac{\delta}{4} < \frac{\delta^*}{4} < \rho.$$

Note that $(\widehat{\beta}_0^{qr}, \widehat{\boldsymbol{\beta}}^{qr})$ is a feasible point of problem (33) by $\widetilde{E} \subseteq E^\star$ and $(\widetilde{\beta}_0^\delta, \widetilde{\boldsymbol{\beta}}^\delta)$ is the optimal solution of problem (33), we have $Q^\delta(\widetilde{\beta}_0^\delta, \widetilde{\boldsymbol{\beta}}^\delta) \le Q^\delta(\widehat{\beta}_0^{qr}, \widehat{\boldsymbol{\beta}}^{qr})$. It then follows from (29) that

$$Q(\widetilde{\beta}_0^\delta, \widetilde{\boldsymbol{\beta}}^\delta) - Q(\widehat{\beta}_0^{qr}, \widehat{\boldsymbol{\beta}}^{qr}) = [Q(\widetilde{\beta}_0^\delta, \widetilde{\boldsymbol{\beta}}^\delta) - Q^\delta(\widetilde{\beta}_0^\delta, \widetilde{\boldsymbol{\beta}}^\delta)] + [Q^\delta(\widetilde{\beta}_0^\delta, \widetilde{\boldsymbol{\beta}}^\delta) - Q^\delta(\widehat{\beta}_0^{qr}, \widehat{\boldsymbol{\beta}}^{qr})] + [Q^\delta(\widehat{\beta}_0^{qr}, \widehat{\boldsymbol{\beta}}^{qr}) - Q(\widehat{\beta}_0^{qr}, \widehat{\boldsymbol{\beta}}^{qr})] < \rho.$$

Algorithm 3 The GCD algorithm for quantile regression

```
1. Initialize $(\widetilde{\beta}_0, \widetilde{\boldsymbol{\beta}})$.
2. Cyclic coordinate descent, for $j = 1, 2, \ldots, p$:
    (a) Compute $r_i = y_i - \widetilde{\beta}_0 - \mathbf{x}_i^\top \widetilde{\boldsymbol{\beta}}$.
    (b) Compute
        $\widetilde{\beta}_j^{\text{new}} = \frac{1}{(\frac{1}{2\delta} + \lambda_2 + \sigma \sum_{i \in D_0} x_{ij}^2)} S\left(\frac{1}{2\delta}\widetilde{\beta}_j + \sum_{i \in D_0} x_{ij}(\theta_i + \sigma(y_i - \widetilde{\beta}_0 - \sum_{t \neq j} x_{it}\widetilde{\beta}_t)) - \frac{1}{n}\sum_{i=1}^{n} H'_{\delta,\tau}(r_i)x_{ij}, \lambda_1\omega_j\right).$
    (c) Set $\widetilde{\beta}_j = \widetilde{\beta}_j^{\text{new}}$.
3. Update the intercept term:
    (a) Compute $r_i = y_i - \widetilde{\beta}_0 - \mathbf{x}_i^\top \widetilde{\boldsymbol{\beta}}$.
    (b) Compute
        $\widetilde{\beta}_0^{\text{new}} = \widetilde{\beta}_0 + \frac{1}{\sigma|D_0| + \frac{1}{2\delta}}\left(\frac{1}{n}\sum_{i=1}^{n} H'_{\delta,\tau}(r_i) + \sum_{i \in D_0} \theta_i + \sigma \sum_{i \in D_0} (y_i - \widetilde{\beta}_0 - \mathbf{x}_i^\top \widetilde{\boldsymbol{\beta}})\right).$
    (c) Set $\widetilde{\beta}_0 = \widetilde{\beta}_0^{\text{new}}$.
4. Update $\boldsymbol{\theta}$, for all $i \in \widetilde{E}$:
    (a) Update $\widetilde{\theta}_i^{\text{new}} = \widetilde{\theta}_i - \sigma\left(y_i - \widetilde{\beta}_0 - \mathbf{x}_i^\top \widetilde{\boldsymbol{\beta}}\right)$.
    (b) Set $\widetilde{\theta}_i = \widetilde{\theta}_i^{\text{new}}$.
5. Repeat steps 2-4 until the convergence of $(\widetilde{\beta}_0, \widetilde{\boldsymbol{\beta}})$.
```

By the definition of $C_{\gamma_0/2}$, we know that $(\widetilde{\beta}_0^\delta, \widetilde{\boldsymbol{\beta}}^\delta) \notin C_{\gamma_0/2}$, therefore $\left|\widetilde{\beta}_0^\delta + \mathbf{x}_i^\top \widetilde{\boldsymbol{\beta}}^\delta - \widehat{\beta}_0^{qr} - \mathbf{x}_i^\top \widehat{\boldsymbol{\beta}}^{qr}\right| < \delta_0/2$ for all i. Furthermore, for any $i \in \widetilde{E}^\delta$,

$$|y_i - \mathbf{x}_i^\top \widehat{\boldsymbol{\beta}}^{qr} - \widehat{\beta}_0^{qr}| \leqslant \left|y_i - \mathbf{x}_i^\top \widetilde{\boldsymbol{\beta}}^\delta - \widetilde{\beta}_0^\delta\right| + \left|(\mathbf{x}_i^\top \widetilde{\boldsymbol{\beta}}^\delta + \widetilde{\beta}_0^\delta) - (\mathbf{x}_i^\top \widehat{\boldsymbol{\beta}}^{qr} + \widehat{\beta}_0^{qr})\right| < \delta + \gamma_0/2 < \gamma_0,$$

by the definition of γ_0, we know that $i \in E^\star$. We conclude that $\widetilde{E} \subseteq \widetilde{E}^\delta \subseteq E^\star$. □

The GCD algorithm, tailored for high-dimensional quantile regression, is comprehensively detailed in Algorithm 3.