Journal of Machine Learning Research 23 (2022) 1-39 Submitted 9/21; Revised 10/22; Published 11/22

# On Regularized Square-root Regression Problems: Distributionally Robust Interpretation and Fast Computations

**Hong T.M. Chu** HONGTMCHU@U.NUS.EDU
*Department of Mathematics*
*National University of Singapore*
*Singapore 119076*

**Kim-Chuan Toh** MATTOHKC@NUS.EDU.SG
*Department of Mathematics, and Institute of Operations Research and Analytics*
*National University of Singapore*
*Singapore 119076*

**Yangjing Zhang** YANGJING.ZHANG@AMSS.AC.CN
*Institute of Applied Mathematics, Academy of Mathematics and Systems Science*
*Chinese Academy of Sciences*
*People's Republic of China 100190*

**Editor:** Francis Bach

## Abstract

Square-root (loss) regularized models have recently become popular in linear regression due to their nice statistical properties. Moreover, some of these models can be interpreted as the distributionally robust optimization counterparts of the traditional least-squares regularized models. In this paper, we give a unified proof to show that any square-root regularized model whose penalty function being the sum of a simple norm and a seminorm can be interpreted as the distributionally robust optimization (DRO) formulation of the corresponding least-squares problem. In particular, the optimal transport cost in the DRO formulation is given by a certain dual form of the penalty. To solve the resulting square-root regularized model whose loss function and penalty function are both nonsmooth, we design a proximal point dual semismooth Newton algorithm and demonstrate its efficiency when the penalty is the sparse group Lasso penalty or the fused Lasso penalty. Extensive experiments demonstrate that our algorithm is highly efficient for solving the square-root sparse group Lasso problems and the square-root fused Lasso problems.

**Keywords:** square-root regularized model, distributionally robust optimization, proximal point algorithm, semismooth Newton, group Lasso

## 1. Introduction

Variable selection in high dimensional feature space has played a pivotal role in contemporary statistical and machine learning studies. Let $(X_1, Y_1), \ldots, (X_N, Y_N)$ be data generated from a linear regression model

$$Y_i = X_i^T \beta_0 + \sigma \epsilon_i, \tag{1}$$

where $X_i = (X_{i1}, \ldots, X_{in})^T \in \mathbb{R}^n$ is the $i$th predictor vector, $Y_i \in \mathbb{R}$ is the corresponding response variable, and $\epsilon_i$'s are independent and identically distributed (i.i.d.) noises such that $\mathrm{E}(\epsilon_i) = 0$ and $\mathrm{E}(\epsilon_i^2) = 1$. The vector $\beta_0 \in \mathbb{R}^n$ is the unknown true regression vector, and $\sigma > 0$ is the unknown noise level. We let $X := (X_1, \ldots, X_N)^T \in \mathbb{R}^{N \times n}$ be the matrix of features and $Y := (Y_1, \ldots, Y_N)^T \in \mathbb{R}^N$ be the response vector. In high dimensional regression problems, the dimension of predictors $n$ is usually much larger than the sample size $N$. With such a large number of predictors, one often prefers to select a smaller subset that is significant and highly correlated to the response for the ease of interpreting the model, as well as to avoid the issue of overfitting. One of the most popular methods for variable selection is the classic Lasso model (Tibshirani, 1996), which minimizes the residual sum of squared errors plus an $\ell_1$ norm penalty term. Further imposing problem-specific assumptions on the structures of the variables, various variants of the classic Lasso model have been proposed. These variants are referred to as the Lasso-type models, and they include for example, the (sparse) group Lasso model (Yuan and Lin, 2006; Friedman et al., 2010), the fused Lasso model (Tibshirani et al., 2005), and the clustered Lasso model (Petry et al., 2011; She, 2010), to mention only a few. To date, large amount of research have been conducted on Lasso-type models, and they have been widely applied in real applications; see for example (Xu et al., 2010b; Muthukrishnan and Rohini, 2016; Yang et al., 2010; Jacob et al., 2009; Angelosante and Giannakis, 2009; Rao et al., 2015; Bazerque et al., 2011), just to name a few. Most Lasso-type models incorporate different regularizers for achieving different underlying sparsity structures in the regression vector and they solve the following generic squared-loss convex optimization problem:

$$\min_{\beta \in \mathbb{R}^n} \quad \left\{ \|Y - X\beta\|^2 + \lambda p(\beta) \right\}, \tag{2}$$

where $\|\cdot\|$ is the Euclidean norm, $p$ is the regularizer, and $\lambda > 0$ is the tuning parameter.

The theoretically optimal values of the tuning parameter $\lambda$ in the Lasso and group Lasso models have been well estimated in (Bickel et al., 2009; Lounici et al., 2011; Meinshausen and Yu, 2009; Zhang and Huang, 2008). However, an outstanding problem is that the choice of $\lambda$ will depend on the unknown noise level $\sigma$, which is typically nontrivial to estimate. An alternative for choosing a suitable $\lambda$ for good generalization performance is by cross validation, but this procedure can be time-consuming in practice. To eliminate the need to know or estimate $\sigma$, Belloni et al. (2011) introduced the square-root Lasso model, and subsequently Bunea et al. (2013) extended it to the square-root group Lasso model. The square-root (loss) regularized model takes the square-root of the residual sum of squared errors as the loss function and any sparsity inducing norm as the regularizer. From now on we refer to the following optimization problem as the square-root regularized model and its solution as the square-root regularized estimator

$$\min_{\beta \in \mathbb{R}^n} \quad \{ \|Y - X\beta\| + \lambda p(\beta) \}. \tag{3}$$

An important general step forward has been made by Stucky and van de Geer (2017), who proved the nice statistical property that the theoretically optimal $\lambda$ for an estimator given by (3) will be independent of the unknown noise level $\sigma$ for any penalty $p$ that is a weakly decomposable norm (van de Geer, 2014, Definition 4.1). Such an estimator is quite general

in that $p$ can be an arbitrary weakly decomposable norm. In fact, it was stated in (Stucky and van de Geer, 2017) that the nice statistical property is applicable for the square-root Lasso, the square-root (sparse) group Lasso, and the square-root SLOPE (Bogdan et al., 2015) estimators. In addition, it has been also shown by Jiang et al. (2021) that the choice of $\lambda$ in the square-root fused Lasso model is independent of $\sigma$. Thus far, we can see that the square-root regularized model (3) is more preferable over the Lasso-type model (2) in terms of their statistical properties.

Recently, there has been renewed interest in the square-root regularized model (3) and its interpretation from the perspective of distributionally robust optimization (DRO). This connection gives a new probabilistic explanation of the penalty level $\lambda$ based on the DRO formulation. A DRO problem aims to find a regression vector $\beta$ that minimizes the worst-case loss over an uncertainty set, namely $\sup_{\mathbb{P}\in\mathcal{U}} \mathrm{E}_{\mathbb{P}}[\ell(X, Y; \beta)]$. Here, $\ell$ denotes a loss function, $\mathrm{E}_{\mathbb{P}}[\cdot]$ denotes the expectation with respect to a probability distribution $\mathbb{P}$ of $(X, Y)$, and $\mathcal{U}$ denotes the uncertainty set of probability measures. The uncertainty set $\mathcal{U}$ specifies prior distributional information about $\mathbb{P}$, and it is usually constructed to include the unknown true distribution with a probabilistic guarantee. Let us now introduce some notation and give the DRO formulation related to (3). We denote the Dirac distribution at $(X_i, Y_i)$ by $\mathbf{1}_{\{(X_i,Y_i)\}}$, the empirical distribution by $\mathbb{P}_N(dx, dy) := \frac{1}{N}\sum_{i=1}^{N} \mathbf{1}_{\{(X_i,Y_i)\}}(dx, dy)$, and the squared-loss function by $\ell(x, y; \beta) := (y - \beta^T x)^2$. Therefore, we have that $\mathrm{E}_{\mathbb{P}_N}[\ell(X, Y; \beta)] = \frac{1}{N}\|Y - X\beta\|^2$, and (3) takes an equivalent form

$$\min_{\beta\in\mathbb{R}^n} \quad \left\{ \sqrt{\mathrm{E}_{\mathbb{P}_N}[\ell(X, Y; \beta)]} + \frac{\lambda}{\sqrt{N}} \cdot p(\beta) \right\}. \tag{4}$$

When $p(\beta) = \|\beta\|_q := (\sum_{i=1}^{n} |\beta_i|^q)^{1/q} \; \forall\, \beta \in \mathbb{R}^n$ and $q \in [1, \infty)$, (4) has the following equivalent DRO representation, as shown in (Blanchet et al., 2019a),

$$\inf_{\beta\in\mathbb{R}^n} \quad \sup_{\mathbb{P}} \left\{ \mathrm{E}_{\mathbb{P}}[\ell(X, Y; \beta)] \,\middle|\, \mathcal{D}_c(\mathbb{P},\, \mathbb{P}_N) \le \frac{\lambda^2}{N} \right\} \tag{5}$$

where $\mathcal{D}_c(\mathbb{P}, \mathbb{Q})$ is an optimal transport cost between two probability measures $\mathbb{P}$ and $\mathbb{Q}$ based on a suitably chosen cost function $c$ (see (9) for the definition of $\mathcal{D}_c$), and $\{\mathbb{P} \,|\, \mathcal{D}_c(\mathbb{P}, \mathbb{P}_N) \le \frac{\lambda^2}{N}\}$ is the uncertainty set centered at the empirical distribution $\mathbb{P}_N$ with radius $\frac{\lambda^2}{N}$. The inner maximization problem of (5) accounts for all the probability measures that are plausible variations of $\mathbb{P}_N$. The problem (5) tries to minimize the worst-case loss and consequently it is likely to perform uniformly well around the empirical distribution. Furthermore, we can interpret from (5) that the regularization parameter $\lambda$ fully quantifies the radius of the uncertainty set $\left\{\mathbb{P} \mid \mathcal{D}_c(\mathbb{P}, \mathbb{P}_N) \le \frac{\lambda^2}{N}\right\}$. In fact, when $p$ is the $\ell_q$ norm, Blanchet et al. (2019a) proved that an associated cost function $c$ defined by the dual norm of the $\ell_q$ norm will give rise to the equivalence between (4) and (5). Besides, Blanchet and Kang (2017) provided a DRO representation for the square-root group Lasso model, i.e., (3) with $p$ being the group Lasso penalty function. The problem (3) also admits a DRO representation when the regularizer is given by $p(\beta) = \|\beta\|_\Lambda := \sqrt{\beta^T \Lambda \beta} \; \forall\, \beta \in \mathbb{R}^n$ with $\Lambda$ being a given symmetric positive definite matrix, as shown in (Blanchet et al., 2019b, Theorem 1). Additionally, the connections between regularization and robust optimization have been

extensively studied in the literature, e.g., (El Ghaoui and Lebret, 1997; Xu et al., 2010a, 2009; Shafieezadeh-Abadeh et al., 2015; Bertsimas and Copenhaver, 2018).

One of our contributions in this paper is to provide a DRO representation for a general class of square-root regularized models where the regularizer can take the general form as the sum of a simple norm and a seminorm. This broad class of regularizers can include most of the popular penalty functions in statistics and machine learning. As a key ingredient in the DRO formulation, the optimal transport cost is given by a certain dual form of the regularizer. Despite the superior statistical properties, square-root regularized models are more complex and challenging to solve compared to their squared-loss counterparts because now the loss function and the penalty function are both nonsmooth. Thus designing efficient algorithms that are capable of solving high-dimensional square-root regularized models is an important task for making these models practically useful. Here we develop a proximal point dual semismooth Newton algorithmic framework for solving a generic square-root regularized model. Specifically, we illustrate how the general framework can be adopted to solve the square-root sparse group Lasso and the square-root fused Lasso models, i.e., for solving (3) when $p$ is one of the following regularizers:

- the sparse group Lasso regularizer (Friedman et al., 2010)

$$p(\beta) = w_1\|\beta\|_1 + w_2 \sum_{l=1}^{g} \omega_l \|\beta_{G_l}\| \ \forall\, \beta \in \mathbb{R}^n, \tag{6}$$

  where $w_1$ and $w_2$ are nonnegative regularization parameters, $\omega_l > 0$ is the weight for the $l$th group, $G_l$'s form a partition of $\{1, \ldots, n\}$, and $\beta_{G_l}$ is the subvector of $\beta$ restricted to $G_l$;

- the fused Lasso regularizer (Tibshirani et al., 2005)

$$p(\beta) = w_1\|\beta\|_1 + w_2 \sum_{i=1}^{n-1} |\beta_i - \beta_{i+1}| \ \forall\, \beta \in \mathbb{R}^n, \tag{7}$$

  where $w_1$ and $w_2$ are nonnegative regularization parameters.

We should mention that although the square-root regularized models of the form (3) enjoy nice statistical properties and insightful DRO interpretation, there is currently no efficient unified algorithmic framework for solving such models, possibly due to challenge posed by the nonsmoothness of the square-root loss function in addition to the nonsmoothness of the penalty function. In fact, existing algorithms are limited to solving special cases such as the square-root Lasso model, and they are not efficient enough for solving large-scale problems. Belloni et al. (2011) reformulated the square-root Lasso model into a second order cone programming (SOCP), and then applied TFOCS (Becker et al., 2011) (first order conic solvers), SDPT3 (Toh et al., 1999; Tütüncü et al., 2003) (interior point methods), and coordinatewise methods for solving the reformulated SOCP. However, this SOCP reformulation will at least double the number of variables—thereby increasing the computational cost substantially, and those off-the-shelf methods are not efficient enough for solving large-scale problems. Additionally, an alternating direction method of multipliers was applied in (Li et al., 2015) for solving the square-root Lasso model. But this approach

may not be efficient for large-scale problems, as can be seen from the numerical experiments in (Tang et al., 2020). Moreover, proximal gradient descent and proximal Newton methods were proposed in (Li et al., 2020) for solving the square-root Lasso model. For solving the square-root group Lasso model, Bunea et al. (2013) proposed a scaled thresholding-based iterative selection procedure. This method highly depends on the choice of the scaling parameter, and it usually requires many iterations for solving large-scale problems, as shown later in our numerical experiments. Moreover, we found that the convergence of the method (Bunea et al., 2013, Theorem 3.1) is based on a regularity condition that depends on the sequence of iterates generated by the method and it may not hold generally. In a recent work, Tang et al. (2020) proposed a semismooth Newton (SSN) based proximal majorization-minimization algorithm for solving nonconvex square-root-loss regression problems, and they demonstrated its efficiency for solving the square-root Lasso model. To the best of our knowledge, an efficient and robust algorithm for solving the square-root regularized model that allows for a general regularizer is still in great demand but not yet available. Inspired by the algorithm of (Tang et al., 2020), we propose in this paper a proximal point dual semismooth Newton algorithm (PPDNA) for solving (3) with regularizer $p$ having the property that its proximal mapping and the associated generalized Jacobian can be computed efficiently. In particular, we implement the PPDNA algorithmic framework to solve the square-root regularized model (3) with the sparse group Lasso regularizer (6) or the fused Lasso regularizer (7). In contrast to the algorithm in (Tang et al., 2020) which does not impose structured sparsity, our algorithm will incorporate structured sparsity imposed by the sparse group Lasso or fused Lasso regularizer. We should add that our PPDNA framework, just like the one in (Tang et al., 2020), is inspired by the highly efficient SSN based proximal point algorithmic (PPA) framework developed solving many squared-loss Lasso-type problems (Li et al., 2018b,c; Lin et al., 2019; Luo et al., 2019; Zhang et al., 2020). The key difference between our current framework and the one developed for a squared-loss Lasso-type problem is that the latter is applied to the dual problem having an essentially smooth strongly convex term in the objective function, but such a property is not present for our square-root regularized models. Fortunately, as we shall see later, the desirable properties of the PPA and SSN, such as the fast convergence speed of the PPA and the ability to exploit second order sparsity in the SSN method, are preserved in our current PPDNA framework when there is no overfitting of the data.

The remaining parts of the paper are organized as follows. In Section 2 we show that the square-root regularized model (3) has an equivalent DRO representation. We design a proximal point dual semismooth Newton algorithm for solving the square-root regularized model (3) in Section 3, and we elucidate the computational details of two cases where the regularizer is chosen to be the sparse group Lasso regularizer (6) or the fused Lasso regularizer (7). We conduct numerical experiments on synthetic and real data sets in Section 4, and give the conclusion in Section 5.

**Notation** Throughout the paper any vector is understood to be a column vector. We denote the inner product of two vectors $x$ and $y$ in $\mathbb{R}^n$ by $\langle x, y\rangle := x^T y$. We denote the Euclidean norm by $\|\cdot\|$ and the unit ball of Euclidean norm by $\mathcal{B} := \{x \in \mathbb{R}^n \mid \|x\| \le 1\}$. For a matrix $X \in \mathbb{R}^{m\times n}$, we denote the operator norm of $X$ by $\|X\| := \inf\{c \mid \|Xv\| \le c\,\|v\|\ \forall v \in \mathbb{R}^n\}$. We denote the elementwise multiplication by $\odot$. We adopt the conventions of extended

arithmetic, whereby $\infty \cdot 0 = 0 \cdot \infty = 0/0 = 0$ and $\infty - \infty = -\infty + \infty = 1/0 = \infty$. For any subset $A \subseteq \mathbb{R}^n$, we denote the cardinality of $A$ as $|A|$. We denote the vector of all ones by $\mathbf{1}$ and the identity matrix by $I$. We denote the vector (or the matrix) of all zeros by $\mathbf{0}$. For a matrix $X \in \mathbb{R}^{m\times n}$ and an index set $G \subseteq \{1, 2, \ldots, n\}$, we denote the range space (resp. null space) of $X$ by $\mathtt{Range}(X)$ (resp. $\mathtt{Null}(X)$) and the submatrix formed by the columns of $X$ corresponding to $G$ by $X_G$. We denote the square diagonal matrix with the elements of vector $v$ on the main diagonal by $\mathtt{Diag}(v)$. For a vector $\beta \in \mathbb{R}^n$, we denote the positive part of $\beta$ by $\beta^+ := \max\{\beta, 0\}$, the vector with the signs of the corresponding elements of $\beta$ by $\mathtt{sign}(\beta)$ (the sign of a real number is 1, 0, or $-1$ if the number is positive, zero, or negative, respectively), the restricted vector of $\beta$ to an index set $G \subseteq \{1, 2, \ldots, n\}$ by $\beta_G$, the support of $\beta$ by $\mathtt{supp}(\beta) := \{i \,|\, \beta_i \neq 0\}$. Given a vector space $\mathcal{V}$, a norm on $\mathcal{V}$ is a nonnegative valued function $p : \mathcal{V} \to \mathbb{R}$ with the following properties: for all $\lambda \in \mathbb{R}$ and $x$, $y \in \mathcal{V}$, (1) (triangle inequality) $p(x + y) \leq p(x) + p(y)$, (2) (absolute homogeneity) $p(\lambda x) = |\lambda| p(x)$, (3) (positive definiteness) $p(x) = 0$ implies $x = 0$. A seminorm on $\mathcal{V}$ is a function $p : \mathcal{V} \to \mathbb{R}$ with the properties (1) and (2) above. For a seminorm $p$ on $\mathbb{R}^n$, we define $p_* : \mathbb{R}^n \to [0, +\infty]$ by $p_*(y) := \sup_x \{\langle y, x\rangle \,|\, p(x) \leq 1\}\ \forall\, y \in \mathbb{R}^n$ (if $p$ is a norm then $p_*$ is also a norm, called its dual norm). For a closed proper convex function $f : \mathbb{R}^n \to (-\infty, +\infty]$, we denote its effective domain by $\mathtt{dom}(f) := \{x \,|\, f(x) < +\infty\}$, its Fenchel conjugate by $f^*(x) := \sup_{y\in\mathbb{R}^n}\{\langle y, x\rangle - f(y)\}$, the proximal mapping of $f$ at $x$ by $\mathtt{prox}_f(x) := \arg\min_y\{f(y) + \frac{1}{2}\|x - y\|^2\}$, and the Moreau envelope (Moreau, 1965; Yosida, 1964) of $f$ at $x$ by $\mathtt{M}_f(x) := \min_y\{f(y) + \frac{1}{2}\|x - y\|^2\}$. The Moreau envelope is continuously differentiable with the gradient $\nabla \mathtt{M}_f(x) = x - \mathtt{prox}_f(x)\ \forall\, x$. The multivariate normal distribution with mean vector $\mu$ and covariance matrix $\Sigma$ is denoted by $\mathcal{N}(\mu, \Sigma)$.

## 2. DRO Formulation of Squared-loss Linear Regression Problems

In this section, we will show that the square-root regularized model (3) is equivalent to a DRO formulation of a squared-loss linear regression problem, namely,

$$\inf_{\beta\in\mathbb{R}^n} \sup_{\mathbb{P}:\, \mathcal{D}_c(\mathbb{P},\mathbb{P}_N)\leq\delta} \mathrm{E}_{\mathbb{P}}[\ell(X, Y; \beta)]. \tag{8}$$

In the above, $c$ is an appropriate optimal transport cost function that will be defined explicitly later, and $\delta$ can be regarded as the radius of the uncertainty set centered at $\mathbb{P}_N$. Later we will show that the penalty parameter $\lambda$ in (3) will fully quantify the radius $\delta$.

### 2.1 Optimal Transport Costs

We introduce in this section some notation and recall the optimal transport cost between probability measures; see (Villani, 2008, Chapter 6) for more details. For any two probability measures $\mathbb{P}$ and $\mathbb{Q}$ in $\mathbb{R}^{n+1}$, $\Pi(\mathbb{P}, \mathbb{Q})$ denotes the set of all joint probability measures on $\mathbb{R}^{n+1} \times \mathbb{R}^{n+1}$ whose marginals are $\mathbb{P}$ and $\mathbb{Q}$. For a given cost function $c : \mathbb{R}^{n+1} \times \mathbb{R}^{n+1} \to [0, \infty]$, where $c(u, v)$ is the cost for transporting one unit of mass from $u$ to $v$, the optimal transport cost between $\mathbb{P}$ and $\mathbb{Q}$ is defined as

$$\mathcal{D}_c(\mathbb{P}, \mathbb{Q}) := \inf_{\pi\in\Pi(\mathbb{P},\mathbb{Q})} \int c(u, v) d\pi(u, v). \tag{9}$$

We assume that $c(u,u) = 0$ for any $u \in \mathbb{R}^{n+1}$. For any nonnegative lower semicontinuous cost function $c$, it is shown in (Villani, 2008, Theorem 4.1) that the infimum in (9) is attainable. Intuitively, one can regard (9) as a kind of distance between two measures $\mathbb{P}$ and $\mathbb{Q}$, but strictly speaking, it is not guaranteed to satisfy the axioms of a distance. However, one can obtain a distance from (9) when the cost is defined in terms of a distance. For example, if the cost function is defined by the $\ell_q$-norm as $c(u,v) = \|u-v\|_q^\rho$, $q \geq 1$, $\rho \geq 1$, then $\mathcal{D}_c(\cdot,\cdot)$ is the well known Wasserstein distance of order $\rho$ (also known as the optimal transport distance or the earth mover's distance). Wasserstein distances and Wasserstein barycenters have recently become very popular and are widely applied in many applications (Yang et al., 2021; Bigot and Klein, 2018; Cuturi and Doucet, 2014; Li and Wang, 2008; Ye et al., 2017; Rabin et al., 2011). In contrast to standard Wasserstein distances, the cost function $c$ in our analysis are more general in that we allow for lower semicontinuous cost functions that may take infinite values. As we will see in Theorem 4, a judicious choice of $c$ will give rise to the equivalence between (3) and (8).

### 2.2 DRO Formulation

In this section, we will give a DRO representation of the square-root regularized model (3) for a broad class of regularizers $p$ expressed as the sum of a simple norm and a seminorm as follows. Let $P : \mathbb{R}^n \to \mathbb{R}$ and $Q : \mathbb{R}^s \to \mathbb{R}$ be two given norms. We consider $p : \mathbb{R}^n \to \mathbb{R}$ defined by

$$p(\beta) := w_1 P(\beta) + w_2 Q(B\beta),\ \beta \in \mathbb{R}^n, \tag{10}$$

where $B \in \mathbb{R}^{s\times n}$ is a given matrix, and $w_1$ and $w_2$ are nonnegative scalars adding up to one. Such a regularizer $p$ and the corresponding dual function $p_*$ are essential ingredients in the definition of the optimal transport cost in the DRO formulation. In Proposition 1, we give a relatively explicit form of $p_*$.

**Proposition 1** *Let $p : \mathbb{R}^n \to \mathbb{R}$ be the function defined in* (10)*. Then the function*

$$p_*(\alpha) := \sup_\beta \{\langle \alpha, \beta\rangle \,|\, p(\beta) \leq 1\},\ \alpha \in \mathbb{R}^n \tag{11}$$

*admits the form*

$$p_*(\alpha) = \inf_{\tilde\alpha\in\mathbb{R}^n,\, \bar\alpha\in\mathbb{R}^s} \sup_{t\in[0,1]} \left\{ t\frac{P_*(\tilde\alpha)}{w_1} + (1-t)\frac{Q_*(\bar\alpha)}{w_2} \,\middle|\, \tilde\alpha + B^T\bar\alpha = \alpha \right\},\ \alpha \in \mathbb{R}^n. \tag{12}$$

Before presenting the proof, we would like to give an explanation of this proposition. By virtue of the convention of extended arithmetic ($0/0 = 0$, $1/0 = \infty$), the formula (12) can also include the special cases with $w_1 = 0$ or $w_2 = 0$, namely,

$$p_*(\alpha) = \begin{cases} P_*(\alpha), & \text{if } w_1 = 1,\ w_2 = 0, \\ \inf\limits_{\bar\alpha\in\mathbb{R}^s} \{Q_*(\bar\alpha) \,|\, B^T\bar\alpha = \alpha\}, & \text{if } w_1 = 0,\ w_2 = 1. \end{cases} \tag{13}$$

The convention of extended arithmetic allows us to have a uniform expression (12) without having to separately write out different cases. Additionally, one can see that $p_*$ in the

second case of (13) will have finite values if and only if $\alpha \in \texttt{Range}(B^T)$. Note that in this case $p(\cdot) = Q(B\cdot)$ may only be a seminorm on $\mathbb{R}^n$ and therefore $p_*$ may take infinite values. We will further characterize the properties of $p_*$ when $p(\cdot) = Q(B\cdot)$ is a seminorm later in Proposition 2.

**Proof** For positive coefficients $w_1 > 0$ and $w_2 > 0$, take an arbitrary $\mathbf{0} \neq \alpha \in \mathbb{R}^n$. We have that

$$\begin{aligned}
&p_*(\alpha) = \sup_{\beta} \{\langle \alpha, \beta\rangle \,|\, p(\beta) \le 1\} \\
&= \sup_{\beta} \inf_{u\ge 0} \{\langle \alpha, \beta\rangle + u(1 - p(\beta))\} \\
&= \inf_{u\ge 0} \sup_{\beta} \{\langle \alpha, \beta\rangle + u - uw_1 P(\beta) - uw_2 Q(B\beta)\} \\
&= \inf_{u\ge 0} \sup_{\beta} \{\langle \alpha, \beta\rangle + u - uw_1 \sup_{\tilde\alpha:\, P_*(\tilde\alpha)\le 1} \langle \beta, \tilde\alpha\rangle - uw_2 \sup_{\bar\alpha:\, Q_*(\bar\alpha)\le 1} \langle B\beta, \bar\alpha\rangle\} \\
&= \inf_{u\ge 0} \{u + \sup_{\beta} \inf_{\tilde\alpha:\, P_*(\tilde\alpha)\le 1,\, \bar\alpha:\, Q_*(\bar\alpha)\le 1} \langle \beta, \alpha - u(w_1\tilde\alpha + w_2 B^T\bar\alpha)\rangle\} \\
&= \inf_{u\ge 0} \{u + \inf_{\tilde\alpha:\, P_*(\tilde\alpha)\le 1,\, \bar\alpha:\, Q_*(\bar\alpha)\le 1} \sup_{\beta} \langle \beta, \alpha - u(w_1\tilde\alpha + w_2 B^T\bar\alpha)\rangle\}.
\end{aligned}$$

The third equality follows from the strong duality theorem (Rockafellar, 1970, Theorem 28.2 and 28.4); the fourth equality applies the definition of a dual norm; and the last equality follows from a standard minimax theorem (Sion, 1958, Corallary 3.3). The supremum over $\beta$ renders that $\alpha - u(w_1\tilde\alpha + w_2 B^T\bar\alpha) = \mathbf{0}$. Therefore, we have that

$$\begin{aligned}
&\inf_{u\ge 0} \{u + \inf_{\tilde\alpha:\, P_*(\tilde\alpha)\le 1,\, \bar\alpha:\, Q_*(\bar\alpha)\le 1} \sup_{\beta} \langle \beta, \alpha - u(w_1\tilde\alpha + w_2 B^T\bar\alpha)\rangle\} \\
&= \inf_{u,\,\tilde\alpha,\,\bar\alpha} \{u \,|\, u \ge 0,\, P_*(\tilde\alpha) \le 1,\, Q_*(\bar\alpha) \le 1,\, u(w_1\tilde\alpha + w_2 B^T\bar\alpha) = \alpha\} \\
&= \inf_{u,\,\tilde\alpha,\,\bar\alpha} \{u \,|\, u > 0,\, P_*(\tilde\alpha) \le uw_1,\, Q_*(\bar\alpha) \le uw_2,\, \tilde\alpha + B^T\bar\alpha = \alpha\} \\
&= \inf_{\tilde\alpha,\,\bar\alpha} \{\max(P_*(\tilde\alpha)/w_1, Q_*(\bar\alpha)/w_2) \,|\, \tilde\alpha + B^T\bar\alpha = \alpha\} \\
&= \inf_{\tilde\alpha,\,\bar\alpha} \sup_{t\in[0,1]} \left\{ t\frac{P_*(\tilde\alpha)}{w_1} + (1-t)\frac{Q_*(\bar\alpha)}{w_2} \,\middle|\, \tilde\alpha + B^T\bar\alpha = \alpha \right\},
\end{aligned}$$

where the second equality holds since $\alpha \neq \mathbf{0}$, $w_1 > 0$, $w_2 > 0$, and we can simply replace the variable $\tilde\alpha$ by $\tilde\alpha/(uw_1)$ and the variable $\bar\alpha$ by $\bar\alpha/(uw_2)$. Therefore, (12) holds for $w_1 > 0$ and $w_2 > 0$.

For the case when $w_1 = 1$ and $w_2 = 0$, the required result follows trivially.

Next, we prove the result for case when $w_1 = 0$ and $w_2 = 1$, where $p(\cdot) = Q(B\,\cdot)$ is a seminorm but it is not necessarily a norm. Take an arbitrary $\mathbf{0} \neq \alpha \in \mathbb{R}^n$. We have that

$$\begin{aligned}
&p_*(\alpha) = \sup_{\beta} \{\langle \alpha, \beta\rangle \,|\, Q(B\beta) \le 1\} = \sup_{\beta} \inf_{u\ge 0} \{\langle \alpha, \beta\rangle + u(1 - Q(B\beta))\} \\
&\le \inf_{u\ge 0} \sup_{\beta} \{\langle \alpha, \beta\rangle + u(1 - Q(B\beta))\} \\
&= \inf_{u\ge 0} \sup_{\beta} \{\langle \alpha, \beta\rangle + u - u \sup_{\bar\alpha:\, Q_*(\bar\alpha)\le 1} \langle B\beta, \bar\alpha\rangle\} \\
&= \inf_{u\ge 0} \{u + \sup_{\beta} \inf_{\bar\alpha:\, Q_*(\bar\alpha)\le 1} \langle \beta, \alpha - uB^T\bar\alpha\rangle\} \\
&= \inf_{u\ge 0} \{u + \inf_{\bar\alpha:\, Q_*(\bar\alpha)\le 1} \sup_{\beta} \langle \beta, \alpha - uB^T\bar\alpha\rangle\}.
\end{aligned}$$

The third inequality follows from the exchange of sup and inf (Rockafellar, 1970, Lemma 36.1); the fourth equality applies the definition of a dual norm; and the last equality follows from a standard minimax theorem (Sion, 1958, Corallary 3.3). The supremum over $\beta$ renders that $\alpha - uB^T\bar{\alpha} = \mathbf{0}$. Therefore, we have that

$$\begin{aligned}&\inf_{u\geq 0}\{u + \inf_{\bar{\alpha}:\, Q_*(\bar{\alpha})\leq 1}\sup_{\beta}\langle\beta, \alpha - uB^T\bar{\alpha}\rangle\} = \inf_{u,\,\bar{\alpha}}\{u \,|\, u\geq 0,\, Q_*(\bar{\alpha})\leq 1,\, uB^T\bar{\alpha} = \alpha\}\\ &= \inf_{u,\,\bar{\alpha}}\{u \,|\, u > 0,\, Q_*(\bar{\alpha})\leq u,\, B^T\bar{\alpha} = \alpha\} = \inf_{\bar{\alpha}}\{Q_*(\bar{\alpha}) \,|\, B^T\bar{\alpha} = \alpha\},\end{aligned}$$

where the second equality holds since $\alpha \neq \mathbf{0}$ and we can simply replace the variable $\bar{\alpha}$ by $\bar{\alpha}/u$.

When $\mathbf{0} \neq \alpha \in \mathtt{Range}(B^T)$, $\inf_{\bar{\alpha}}\{Q_*(\bar{\alpha}) \,|\, B^T\bar{\alpha} = \alpha\}$ is finite. This, together with the strong duality theorem (Rockafellar, 1970, Theorem 28.2 and 28.4), implies that equality holds in the above inequality $(*)$. Namely, $p_*(\alpha) = \inf_{\bar{\alpha}}\{Q_*(\bar{\alpha}) \,|\, B^T\bar{\alpha} = \alpha\} < +\infty$ for $\mathbf{0} \neq \alpha \in \mathtt{Range}(B^T)$.

When $\mathbf{0} \neq \alpha \notin \mathtt{Range}(B^T)$, $\inf_{\bar{\alpha}}\{Q_*(\bar{\alpha}) \,|\, B^T\bar{\alpha} = \alpha\}$ is infinite. We can have an orthogonal decomposition of $\alpha$: $\alpha = \alpha_r + \alpha_n$, $\alpha_r \in \mathtt{Range}(B^T)$, $\mathbf{0} \neq \alpha_n \in \mathtt{Null}(B)$. Note that $p(\alpha_n) = 0$, and by (11) it holds that $p_*(\alpha) \geq \langle\alpha, k\alpha_n\rangle = k\|\alpha_n\|^2 \to +\infty$, as $k \to +\infty$. Therefore, $p_*(\alpha) = \inf_{\bar{\alpha}}\{Q_*(\bar{\alpha}) \,|\, B^T\bar{\alpha} = \alpha\} = +\infty$ for $\mathbf{0} \neq \alpha \notin \mathtt{Range}(B^T)$.

Together with $p_*(0) = 0$, the proof is completed. ■

For a seminorm $p$, we specify some properties of $p$ and $p_*$. Note that if $p$ is a norm on $\mathbb{R}^n$, then $p_*$ is also a norm on $\mathbb{R}^n$ and the following properties (b) and (c) are standard.

**Proposition 2** *Let $B \in \mathbb{R}^{s\times n}$ be a given matrix, $Q : \mathbb{R}^s \to \mathbb{R}$ be a norm, and $p : \mathbb{R}^n \to \mathbb{R}$ be defined by*

$$p(\beta) = Q(B\beta),\ \beta \in \mathbb{R}^n. \tag{14}$$

*Then $p_*$ defined by* (11) *admits the expression*

$$p_*(\alpha) = \inf_{\bar{\alpha}\in\mathbb{R}^s}\{Q_*(\bar{\alpha}) \,|\, B^T\bar{\alpha} = \alpha\},\ \alpha \in \mathbb{R}^n, \tag{15}$$

*and the following holds:*

(a) $\mathtt{dom}(p_*) = \mathtt{Range}(B^T)$ *and* $p_* : \mathtt{Range}(B^T) \to \mathbb{R}$ *is a norm on the vector space* $\mathtt{Range}(B^T)$*;*

(b) $\alpha^T\beta \leq p(\beta)p_*(\alpha)\ \forall\, \alpha, \beta \in \mathbb{R}^n$*;*

(c) $p(\beta) = \sup_\alpha\{\alpha^T\beta \,|\, p_*(\alpha) \leq 1\}\ \forall\, \beta \in \mathbb{R}^n$*. Moreover, this supremum is achievable. Namely, there exists $\alpha_\beta \in \mathtt{Range}(B^T)$ such that $p_*(\alpha_\beta) = 1$ and $p(\beta) = \alpha_\beta^T\beta$.*

**Proof** (15) follows from Proposition 1, and it implies that $\mathtt{dom}(p_*) = \mathtt{Range}(B^T)$. First, we prove that $p_* : \mathtt{Range}(B^T) \to \mathbb{R}$ is a norm on $\mathtt{Range}(B^T)$. The triangle inequality is inherited from that of $Q_*$: $p_*(\alpha+\beta) = \inf_\gamma\{Q_*(\gamma) \,|\, B^T\gamma = \alpha+\beta\} \leq \inf_{\bar{\alpha},\bar{\beta}}\{Q_*(\bar{\alpha}+\bar{\beta}) \,|\, B^T\bar{\alpha} = \alpha, B^T\bar{\beta} = \beta\} \leq \inf_{\bar{\alpha}}\{Q_*(\bar{\alpha}) \,|\, B^T\bar{\alpha} = \alpha\} + \inf_{\bar{\beta}}\{Q_*(\bar{\beta}) \,|\, B^T\bar{\beta} = \beta\} = p_*(\alpha) + p_*(\beta)$. The absolute homogeneity is obvious from that of $Q_*$. If $p_*(\alpha) = 0$, then by the

property of infimum there exist a sequence $\bar{\alpha}_k, k = 1, 2, \ldots$ such that $B^T\bar{\alpha}_k = \alpha$ and $Q_*(\bar{\alpha}_k) < 1/k$. Since $Q_*$ is a norm on $\mathbb{R}^s$, the latter further implies that $\bar{\alpha}_k \to 0$ and hence $\alpha = 0$. Therefore, $p_*$ is a norm on $\texttt{Range}(B^T)$.

Second, we prove (b). For any $\alpha$, $\beta$, and $\bar{\alpha}$ satisfying $B^T\bar{\alpha} = \alpha$, we have the inequality $\alpha^T\beta = (B\beta)^T\bar{\alpha} \le Q(B\beta)Q_*(\bar{\alpha})$ since $Q$ is a norm. We can deduce (b) by taking infimum over $\bar{\alpha}$.

Third, we prove (c). On one hand, it follows from (b) that $\sup_\alpha\{\alpha^T\beta \,|\, p_*(\alpha) \le 1\} \le \sup_\alpha\{p(\beta)p_*(\alpha) \,|\, p_*(\alpha) \le 1\} \le p(\beta)\ \forall\, \beta$. On the other hand, we have that $\sup_\alpha\{\alpha^T\beta \,|\, p_*(\alpha) \le 1\} = \sup_{\alpha\in\texttt{Range}(B^T)}\{\alpha^T\beta \,|\, \inf_{\bar{\alpha}}\{Q_*(\bar{\alpha}) \,|\, B^T\bar{\alpha} = \alpha\} \le 1\} = \sup_\gamma\{(B\beta)^T\gamma \,|\, \inf_{\bar{\alpha}}\{Q_*(\bar{\alpha}) \,|\, B^T\bar{\alpha} = B^T\gamma\} \le 1\} \ge \sup_\gamma\{(B\beta)^T\gamma \,|\, Q_*(\gamma) \le 1\} = Q(B\beta) = p(\beta)$, where the second last equality follows from the properties of the norm $Q$ and its dual norm $Q_*$. Moreover, from (a) we know that $p_*$ is a norm on $\texttt{Range}(B^T)$ and therefore the set $\{\alpha \,|\, p_*(\alpha) \le 1\}$ is compact. Then it is easy to show that this supremum is achievable at the boundary. The proof is completed. ■

Proposition 2 includes a similar result in (Maurer and Pontil, 2012) about the operator norm $\|\beta\|_{\mathcal{M}^*} := \sup_{1\le l\le g}\left\{\|M_{(l)}\beta\|\right\}\ \forall\, \beta \in \mathbb{R}^n$ and its dual form, with $\mathcal{M} := \{M_{(l)}\}_{1\le l\le g}$ being a set of symmetric matrices $M_{(l)} \in \mathbb{S}^n$. See Appendix A for details.

The next proposition about strong duality is a direct application of (Blanchet and Murthy, 2019, Theorem 1). It shows that the inner maximization in the DRO problem (8) has a nice univariate dual problem.

**Proposition 3** *(Blanchet et al., 2019a, Proposition 1) Let $c : \mathbb{R}^{n+1}\times\mathbb{R}^{n+1} \to [0, +\infty]$ be a lower semicontinuous cost function satisfying $c((x, y), (x', y')) = 0$ whenever $(x, y) = (x', y')$. For $\gamma \ge 0$ and a loss function $\ell(x, y; \beta)$ that is upper semicontinuous in $(x, y)$ for each $\beta$, define*

$$\phi_\gamma(X_i, Y_i; \beta) := \sup_{u\in\mathbb{R}^n,\, v\in\mathbb{R}} \left\{\ell(u, v; \beta) - \gamma c((u, v), (X_i, Y_i))\right\}.$$

*Then*

$$\sup_{\mathbb{P}:\, \mathcal{D}_c(\mathbb{P},\mathbb{P}_N)\le\delta} \mathrm{E}_{\mathbb{P}}[\ell(X, Y; \beta)] = \inf_{\gamma\ge 0}\left\{\gamma\delta + \frac{1}{N}\sum_{i=1}^N \phi_\gamma(X_i, Y_i; \beta)\right\}.$$

*Consequently, the DRO problem* (8) *reduces to*

$$\inf_{\beta\in\mathbb{R}^n}\ \sup_{\mathbb{P}:\, \mathcal{D}_c(\mathbb{P},\mathbb{P}_N)\le\delta} \mathrm{E}_{\mathbb{P}}[\ell(X, Y; \beta)] = \inf_{\beta\in\mathbb{R}^n}\ \inf_{\gamma\ge 0}\left\{\gamma\delta + \frac{1}{N}\sum_{i=1}^N \phi_\gamma(X_i, Y_i; \beta)\right\}.$$

Based on Proposition 3, we will prove in the next main theorem the equivalence between the square-root regularized model (3) and the DRO formulation (8) by finding an explicit form of $\phi_\gamma$. This theorem not only unifies existing results (Blanchet and Kang, 2017; Blanchet et al., 2019a) but also include a broader class of regularizers; it is applicable for any regularizer in the additive form expressed in (10).

**Theorem 4** *Consider the squared-loss function $\ell(x, y; \beta) = (y - \beta^T x)^2$ and a regularizer $p$ of the form* (10)*. Let the cost function $c : \mathbb{R}^{n+1}\times\mathbb{R}^{n+1} \to [0, +\infty]$ be defined by*

$$c((u, v), (x, y)) := \begin{cases} (p_*(u - x))^2, & \text{if } v = y,\\ +\infty, & \text{otherwise,}\end{cases}$$

*and the associated optimal transport cost $\mathcal{D}_c(\cdot,\cdot)$ be defined by* (9)*. Then it holds that*

$$\inf_{\beta\in\mathbb{R}^n}\sup_{\mathbb{P}:\,\mathcal{D}_c(\mathbb{P},\mathbb{P}_N)\le\delta}\mathrm{E}_{\mathbb{P}}[\ell(X,Y;\beta)]=\frac{1}{N}\cdot\inf_{\beta\in\mathbb{R}^n}\left\{\|Y-X\beta\|+\sqrt{\delta N}\,p(\beta)\right\}^2.$$

**Proof** Since $c$ is lower semicontinuous with $c((u,v),(u,v))=0$ for any $(u,v)$, and $\ell(x,y;\beta)$ is upper semicontinuous in $(x,y)$ for each $\beta$, based on Proposition 3, we can prove the required result via finding an explicit form of $\phi_\gamma$. Take $\gamma\ge 0$ and $\beta\in\mathbb{R}^n$ arbitrarily. By the definitions of $\phi_\gamma$, $c$, and $\ell$, we have that

$$\phi_\gamma(X_i,Y_i;\beta)=\sup_u\left\{(Y_i-\beta^Tu)^2-\gamma(p_*(u-X_i))^2\,|\,u-X_i\in\mathtt{dom}(p_*)\right\}.$$

For notational simplicity, we denote $\Delta:=u-X_i$ and $Z_i=Y_i-\beta^TX_i$. Then it holds that

$$\begin{aligned}\phi_\gamma(X_i,Y_i;\beta)&=\sup_{\Delta\in\mathtt{dom}(p_*)}\left\{Z_i^2-2Z_i\beta^T\Delta+(\beta^T\Delta)^2-\gamma(p_*(\Delta))^2\right\}&(16)\\&\le Z_i^2+\sup_{\Delta\in\mathtt{dom}(p_*)}\left\{2|Z_i|p(\beta)p_*(\Delta)+\left((p(\beta))^2-\gamma\right)(p_*(\Delta))^2\right\}.&(17)\end{aligned}$$

By the property of a norm and Proposition 2, there exists $\alpha_\beta\in\mathtt{dom}(p_*)$ such that $p_*(\alpha_\beta)=1$ and $p(\beta)=\alpha_\beta^T\beta$.

If $(p(\beta))^2<\gamma$, then the quadratic function in terms of $p_*(\Delta)$ in the above supremum problem (17) is bounded by the value $\frac{Z_i^2(p(\beta))^2}{\gamma-(p(\beta))^2}$ at the stationary point $p_*(\Delta)=\frac{|Z_i|p(\beta)}{\gamma-(p(\beta))^2}$. That is, $\phi_\gamma(X_i,Y_i;\beta)\le Z_i^2+\frac{Z_i^2(p(\beta))^2}{\gamma-(p(\beta))^2}$. It is easy to check that this equality is achievable when one substitutes $\Delta=\frac{-Z_ip(\beta)}{\gamma-(p(\beta))^2}\alpha_\beta$ into (16).

If $(p(\beta))^2>\gamma$, by taking $\Delta=k\alpha_\beta$, $k>0$ in (16), then it holds that $\phi_\gamma(X_i,Y_i;\beta)\ge Z_i^2-2kZ_ip(\beta)+k^2(p(\beta)-\gamma)$. Since $k$ can be arbitrarily large, we deduce that $\phi_\gamma(X_i,Y_i;\beta)=+\infty$.

Lastly, we consider the case where $(p(\beta))^2=\gamma$. If $Z_ip(\beta)=0$, we have that $\phi_\gamma(X_i,Y_i;\beta)=Z_i^2$ since (16) implies that $\phi_\gamma(X_i,Y_i;\beta)\ge Z_i^2$ by taking $\Delta=\mathbf{0}$ and (17) implies that $\phi_\gamma(X_i,Y_i;\beta)\le Z_i^2$. If $Z_ip(\beta)\ne 0$, by taking $\Delta=-kZ_ip(\beta)\alpha_\beta$ with $k>0$ in (16), then it holds that $\phi_\gamma(X_i,Y_i;\beta)\ge Z_i^2+2kZ_i^2(p(\beta))^2$. Since $k$ can be arbitrarily large, we deduce that $\phi_\gamma(X_i,Y_i;\beta)=+\infty$. Therefore, when $(p(\beta))^2=\gamma$, we have that $\phi_\gamma(X_i,Y_i;\beta)=\begin{cases}Z_i^2, & \text{if } Z_ip(\beta)=0,\\ +\infty, & \text{if } Z_ip(\beta)\ne 0.\end{cases}$

In summary, we can have a unified expression if we adopt the conventions of extended arithmetic given in the paragraph on notation, i.e.,

$$\phi_\gamma(X_i,Y_i;\beta)=\begin{cases}Z_i^2+\dfrac{Z_i^2(p(\beta))^2}{\gamma-(p(\beta))^2}, & \text{if } (p(\beta))^2<\gamma, \text{ or } (p(\beta))^2=\gamma \text{ and } Z_ip(\beta)=0,\\ +\infty, & \text{if } (p(\beta))^2>\gamma, \text{ or } (p(\beta))^2=\gamma \text{ and } Z_ip(\beta)\ne 0.\end{cases}\tag{18}$$

Next, we consider the problem

$$\inf_{\gamma\ge 0}\left\{\gamma\delta+\frac{1}{N}\sum_{i=1}^N\phi_\gamma(X_i,Y_i;\beta)\right\}.\tag{19}$$

If $\|Y - X\beta\| = 0$ (namely, $Z_i = 0, \forall\, i$), or $p(\beta) = 0$, then it follows from (18) that (19) reduces to

$$\inf_{\gamma \geq (p(\beta))^2} \left\{ \gamma\delta + \frac{1}{N}\sum_{i=1}^{N} Z_i^2 \right\} = \frac{1}{N}\|Y - X\beta\|^2 + \delta(p(\beta))^2 = \left( \frac{1}{\sqrt{N}}\|Y - X\beta\| + \sqrt{\delta}p(\beta) \right)^2.$$

If $\|Y - X\beta\| \neq 0$ and $p(\beta) \neq 0$, then it follows from (18) that (19) reduces to

$$\inf_{\gamma > (p(\beta))^2} \left\{ \gamma\delta + \frac{1}{N}\sum_{i=1}^{N} \left( Z_i^2 + \frac{Z_i^2 (p(\beta))^2}{\gamma - (p(\beta))^2} \right) \right\} = \left( \frac{1}{\sqrt{N}}\|Y - X\beta\| + \sqrt{\delta}p(\beta) \right)^2,$$

where the minimal value is achieved at $\gamma = \frac{1}{\sqrt{N\delta}}\|Y - X\beta\|p(\beta) + (p(\beta))^2$. Thus far, we have that $\inf_{\gamma \geq 0} \left\{ \gamma\delta + \frac{1}{N}\sum_{i=1}^{N} \phi_\gamma(X_i, Y_i; \beta) \right\} = \left( \frac{1}{\sqrt{N}}\|Y - X\beta\| + \sqrt{\delta}p(\beta) \right)^2$. We further take minimization over $\beta$ on both sides. This, together with Proposition 3, completes the proof. ■

Theorem 4 is applicable for any regularizer of the additive form (10). This form may be a seminorm when $p(\cdot) = Q(B\cdot)$. In this case, one may take into consideration the effective domain of $p_*$, and then the cost function can be written as

$$c((u,v),(x,y)) := \begin{cases} (p_*(u-x))^2, & \text{if } v = y \text{ and } u - x \in \mathtt{Range}(B^T), \\ +\infty, & \text{otherwise.} \end{cases}$$

Our equivalence result in Theorem 4 can cover a broad class of regularizers composed of a norm and a seminorm as in (10). We can obtain from Theorem 4 that the following square-root regularized estimators have equivalent DRO formulations:

- the square-root Lasso estimator (Belloni et al., 2011); a solution of (3) with $p(\beta) = \|\beta\|_1$, $\beta \in \mathbb{R}^n$;
- the square-root sparse group Lasso estimator (Stucky and van de Geer, 2017, Section 4.4); a solution of (3) with $p$ given by (6);
- the square-root SLOPE estimator (Stucky and van de Geer, 2017, Section 4.3); a solution of (3) with $p$ being the following weighted and sorted $\ell_1$ norm with respect to a nonincreasing sequence of weights $\omega_1 \geq \cdots \geq \omega_n > 0$: $p(\beta) = \sum_{i=1}^{n} \omega_i |\beta_i^{\downarrow}|$, where $\beta^{\downarrow}$ is a vector obtained from $\beta$ by sorting its entries in nonincreasing order of magnitude. This $p$ was shown to be a norm in (Zeng and Figueiredo, 2014, Lemma 2);
- the square-root fused Lasso estimator (Jiang et al., 2021); a solution of (3) with $p$ being given by (7);
- the SSASR estimator (Xie and Yang, 2020); a solution of (3) with $p$ being given by $p(\beta) = w_1 \sum_{l=1}^{g} \sqrt{|G_l|}\|\beta_{G_l}\| + w_2 \sum_{l=1}^{g} \sqrt{|G_l|}\|B_l\beta_{G_l}\|$, where the matrix $B_l \in \mathbb{R}^{(|G_l|-1)\times|G_l|}$ is defined as $B_l x = (x_1 - x_2, \ldots, x_{|G_l|-1} - x_{|G_l|})^T \ \forall\, x \in \mathbb{R}^{|G_l|}$ if $|G_l| \geq 2$; and $B_l = 0$ if $|G_l| = 1$.

## 3. A Proximal Point Dual Semismooth Newton Algorithm for Solving the Square-root Regularized Problem

In this section we aim to design a fast algorithm to solve the square-root regularized problem (3). It can be rewritten as follows with an auxiliary vector $y \in \mathbb{R}^N$:

$$\min_{\beta\in\mathbb{R}^n,\, y\in\mathbb{R}^N} \{\|y\| + \lambda p(\beta) \,|\, X\beta - Y = y\}. \tag{20}$$

The dual problem of (20) is given by

$$-\min_{u\in\mathbb{R}^N} \big\{\langle Y, u\rangle + (\lambda p)^*(X^T u) + \delta_{\mathcal{B}}(u)\big\} \tag{21}$$

where $\mathcal{B}$ is the unit Euclidean ball.

Compared with the Lasso-type problem (2), the square-root regularized problem is more challenging to solve since both the loss function $\|\cdot\|$ and the regularizer $p$ are nonsmooth. We aim to use the framework of a proximal point algorithm (PPA) (Rockafellar, 1976) for solving (20). Given two sequences of positive parameters $\{\sigma_k\}$ and $\{\tau_k\}$ such that $\sigma_k \downarrow \underline{\sigma} > 0$ and $\tau_k \downarrow \underline{\tau} > 0$, and an initial point $\big(\beta^0, y^0\big) \in \mathbb{R}^n \times \mathbb{R}^N$, the PPA for solving (20) generates a sequence $\big\{\big(\beta^{k+1}, y^{k+1}\big)\big\}$ via

$$(\beta^{k+1}, y^{k+1}) \approx \mathcal{P}_k(\beta^k, y^k) := \underset{\beta\in\mathbb{R}^n,\, y\in\mathbb{R}^N}{\arg\min} \left\{ \begin{array}{c} \|y\| + \lambda p(\beta) \\ +\frac{\sigma_k}{2}\|\beta - \beta^k\|^2 + \frac{\tau_k}{2}\|y - y^k\|^2 \end{array} \,\middle|\, X\beta - Y = y \right\}. \tag{22}$$

**Algorithm 1:** Proximal point algorithm for solving (20)

**Input** Data $X \in \mathbb{R}^{N\times n}$, $Y \in \mathbb{R}^N$, a penalty parameter $\lambda > 0$, and a regularizer $p$.
**Initialize** $\beta^0 = \mathbf{0} \in \mathbb{R}^n$, $y^0 = -Y$, $\sigma_0 = \tau_0 = 1$.
**while** *the termination criterion is not met,* **do**
  **Step 1.** Update $\beta^{k+1}$ and $y^{k+1}$ by solving (22).
  **Step 2.** Update $\sigma_{k+1}$ and $\tau_{k+1}$. Set $k \leftarrow k + 1$.
**end**
**Output** $\beta^k$ and $y^k$.

We use the standard criterion by Rockafellar (1976) for controlling the inexactness when solving (22) in Algorithm 1:

$$\|(\beta^{k+1}, y^{k+1}) - \mathcal{P}_k(\beta^k, y^k)\| \le \varepsilon_k,\ \varepsilon_k > 0,\ \sum_{k\ge 0} \varepsilon_k < +\infty.$$

The global convergence of Algorihm 1 follows from (Rockafellar, 1976) directly. A key difficulty in Algorithm 1 is how to solve (22) efficiently. Given $\sigma_k > 0$, $\tau_k > 0$, $\beta^k \in \mathbb{R}^n$, and $y^k \in \mathbb{R}^N$, we recall the subproblem (22) given by

$$\begin{array}{ll} \min\limits_{\beta,y} & \|y\| + \lambda p(\beta) + \dfrac{\sigma_k}{2}\|\beta - \beta^k\|^2 + \dfrac{\tau_k}{2}\|y - y^k\|^2 \\ \text{s.t.} & X\beta - Y = y. \end{array} \tag{23}$$

The Lagrangian function associated with (23) is given by

$$\begin{aligned}\mathcal{L}(\beta, y; u) \quad &= \|y\| + \frac{\tau_k}{2}\left\|y - y^k - \tau_k^{-1}u\right\|^2 - \left\langle y^k, u\right\rangle - \frac{1}{2\tau_k}\|u\|^2 - \langle Y, u\rangle + \lambda p(\beta) \\ &\quad + \tfrac{\sigma_k}{2}\left\|\beta - \beta^k + \sigma_k^{-1}X^T u\right\|^2 + \left\langle \beta^k, X^T u\right\rangle - \tfrac{1}{2\sigma_k}\left\|X^T u\right\|^2,\end{aligned}$$

for $(\beta, y, u) \in \mathbb{R}^n \times \mathbb{R}^N \times \mathbb{R}^N$. By some simple manipulations, we can obtain that the dual problem of (23), i.e., $\max_u \min_{\beta,y} \mathcal{L}(\beta, y; u)$, is given by

$$\begin{aligned}-\min_{u\in\mathbb{R}^N} \quad \Psi(u) := &\frac{1}{2\tau_k}\|u\|^2 + \frac{1}{2\sigma_k}\|X^T u\|^2 - \langle u, X\beta^k - y^k - Y\rangle \\ &- \sigma_k \mathtt{M}_{\frac{\lambda}{\sigma_k}p}\left(\beta^k - \sigma_k^{-1}X^T u\right) - \tau_k \mathtt{M}_{\frac{1}{\tau_k}\|\cdot\|}\left(y^k + \tau_k^{-1}u\right).\end{aligned} \tag{24}$$

It follows from the properties of Moreau envelope that the dual objective function $\Psi$ is continuously differentiable and convex. Moreover, the gradient of $\Psi$ is given by

$$\nabla\Psi(u) = -X\mathtt{prox}_{\frac{\lambda}{\sigma_k}p}\left(\beta^k - \sigma_k^{-1}X^T u\right) + \mathtt{prox}_{\frac{1}{\tau_k}\|\cdot\|}\left(y^k + \tau_k^{-1}u\right) + Y,\ u \in \mathbb{R}^N. \tag{25}$$

Thus the problem (24) is an unconstrained smooth convex minimization problem. We let a dual optimal solution be $\bar{u} \in \arg\min \Psi(u)$. Then the optimal solution $(\bar{\beta}, \bar{y})$ to (23) can be computed by

$$\bar{\beta} = \mathtt{prox}_{\frac{\lambda}{\sigma_k}p}\left(\beta^k - \sigma_k^{-1}X^T\bar{u}\right), \quad \bar{y} = \mathtt{prox}_{\frac{1}{\tau_k}\|\cdot\|}\left(y^k + \tau_k^{-1}\bar{u}\right).$$

Due to the favourable property of the dual problem (24), we propose to solve (23) via its dual. In particular, the optimal solution of (24) is nothing but the solution of the nonlinear system $\nabla\Psi(u) = 0$, $u \in \mathbb{R}^N$. The latter can be solved by a semismooth Newton (SSN) method. In order to apply the SSN method, we have to characterize a certain generalized Jacobian of $\nabla\Psi(\cdot)$, which in turns depends on the generalized Jacobian of $\mathtt{prox}_{\frac{\lambda}{\sigma_k}p}(\cdot)$. Since the proximal mappings $\mathtt{prox}_{\frac{\lambda}{\sigma_k}p}(\cdot)$ and $\mathtt{prox}_{\frac{1}{\tau_k}\|\cdot\|}(\cdot)$ are Lipschitz continuous, the following multifunction, which is considered as a generalized Jacobian of $\nabla\Psi(u)$, is well defined:

$$\widehat{\partial}(\nabla\Psi)(u) = \left\{ \sigma_k^{-1}XUX^T + \tau_k^{-1}V \;\middle|\; \begin{array}{l} U \in \partial\mathtt{prox}_{\frac{\lambda}{\sigma_k}p}\left(\beta^k - \sigma_k^{-1}X^T u\right), \\ V \in \partial\mathtt{prox}_{\frac{1}{\tau_k}\|\cdot\|}\left(y^k + \tau_k^{-1}u\right) \end{array} \right\}.$$

Once an element $H \in \widehat{\partial}(\nabla\Psi)(u)$ can be constructed explicitly for any given $u$, the SSN method can be implemented as follows.

**Algorithm 2:** Semismooth Newton method for solving (24)

**Input** Data $X \in \mathbb{R}^{N\times n}$, $Y \in \mathbb{R}^N$, a penalty parameter $\lambda > 0$, a regularizer $p$. $\beta^k \in \mathbb{R}^n$, $y^k \in \mathbb{R}^N$, $\sigma_k > 0$, $\tau_k > 0$. $\eta \in (0,1)$, $\varrho \in (0,1]$. $\rho \in (0,1)$, $\mu \in (0, 0.5)$. $\texttt{tol} > 0$.

**Initialize** $u^0 = \mathbf{0} \in \mathbb{R}^N$, $f^0 = \nabla\Psi(u^0)$ by (25), $j = 0$.

**while** $\left\|f^j\right\| > \texttt{tol}$, **do**

**Step 1.** Find an element $H^j \in \widehat{\partial}(\nabla\Psi)(u^j)$, and then find an approximate solution $d^j$ to the linear system

$$H^j d = -f^j$$

such that $\|H^j d + f^j\| \le \min(\eta, \|f^j\|^{1+\varrho})$.

**Step 2.** Find a step size $\alpha_j = \rho^{m_j}$, where $m_j$ is the smallest nonnegative integer $m$ for which

$$\Psi(u^j + \rho^{m_j} d^j) \le \Psi(u^j) + \mu\rho^{m_j}\langle\nabla\Psi(u^j), d^j\rangle.$$

**Step 3.** Update $u^{j+1} = u^j + \alpha_j d^j$ and compute $f^{j+1} = \nabla\Psi(u^j)$ by (25).

**Step 4.** Set $j \leftarrow j + 1$.

**end**

**Output** $u^j$.

We can show that if the optimal solution $(\bar\beta, \bar y)$ to (23) does not overfit, i.e, $\bar y = X\bar\beta - Y \neq \mathbf{0}$, then at the optimal solution $\bar u$ to (24), the generalized Jacobian $\partial\texttt{prox}_{\frac{1}{\tau_k}\|\cdot\|}\left(y^k + \tau_k^{-1}\bar u\right)$ is a singleton and the element is positive definite. But as the proof follows a similar argument to the one in (Tang et al., 2020, Proposition 12), we omit it here. The above property is crucial to guarantee the fast convergence of the SSN method for solving (24). We state the standard convergence result of the SSN method (Algorithm 2) without proof.

**Theorem 5** *Let $(\bar\beta, \bar y)$ be the optimal solution to the problem (23). Assume that the optimal solution does not overfit the data, i.e., $\bar y = X\bar\beta - Y \neq \mathbf{0}$. Then the sequence $\{u^j\}$ generated by Algorithm 2 converges globally to the unique solution $\bar u$ of (24). Furthermore, the local rate of convergence is of order $1 + \varrho$, with $\varrho \in (0,1]$ given in Algorithm 2, i.e., for all $j$ sufficiently large, $\|u^{j+1} - \bar u\| = \mathcal{O}(\|u^j - \bar u\|^{1+\varrho})$.*

From now on, we restrict our discussions to the case where the regularizer $p$ is either the sparse group Lasso regularizer (6) or the fused Lasso regularizer (7). We will illustrate the explicit form of an element $H \in \widehat{\partial}(\nabla\Psi)(\cdot)$ based on (Li et al., 2018b; Zhang et al., 2020). It is worth noting that the matrix $H$ given below has (structured) sparsity inherited from that of matrices in the generalized Jacobian $\partial\texttt{prox}_p(\cdot)$, and it is known as second order sparsity. The second order sparsity will reduce substantially the computational cost in the SSN method, as demonstrated in (Li et al., 2018c,b; Lin et al., 2019; Luo et al., 2019; Zhang et al., 2020).

**Sparse group Lasso regularizer.** We will first illustrate the construction of an element in $\widehat{\partial}(\nabla\Psi)(\cdot)$ when $p$ is the sparse group Lasso regularizer given by (6). Given positive constants $\kappa_1, \kappa_2$ and $\beta \in \mathbb{R}^n$, it is well known that for any $\beta$

$$\mathtt{prox}_{\kappa_1\|\cdot\|_1}(\beta) = (|\beta| - \kappa_1\mathbf{1})^+ \odot \mathtt{sign}(\beta), \qquad \mathtt{prox}_{\kappa_2\|\cdot\|}(\beta) = \begin{cases} \left(1 - \frac{\kappa_2}{\|\beta\|}\right)^+ \beta, & \text{if } \beta \neq 0, \\ \mathbf{0}, & \text{if } \beta = 0. \end{cases}$$

We can construct a matrix $\boldsymbol{U}_{\kappa_1}(\beta)$ in $\partial\mathtt{prox}_{\kappa_1\|\cdot\|_1}(\beta)$ and a matrix $\boldsymbol{V}_{\kappa_2}(\beta)$ in $\partial\mathtt{prox}_{\kappa_2\|\cdot\|}(\beta)$ for any $\beta$ respectively as follows

$$\begin{aligned} \partial\mathtt{prox}_{\kappa_1\|\cdot\|_1}(\beta) &\ni \boldsymbol{U}_{\kappa_1}(\beta) := \mathtt{Diag}(\boldsymbol{v}), \\ \partial\mathtt{prox}_{\kappa_2\|\cdot\|}(\beta) &\ni \boldsymbol{V}_{\kappa_2}(\beta) := \begin{cases} \left(1 - \frac{\kappa_2}{\|\beta\|}\right) I + \frac{\kappa_2}{\|\beta\|^3}\beta\beta^T, & \text{if } \|\beta\| > \kappa_2, \\ \mathbf{0}, & \text{if } \|\beta\| \leq \kappa_2, \end{cases} \end{aligned}$$

where $\boldsymbol{v} \in \mathbb{R}^n$ is defined by $\boldsymbol{v}_i = 1$ if $|\beta_i| > \kappa_1$, and $\boldsymbol{v}_i = 0$ otherwise. Let $\tilde{\beta} := \beta^k - \sigma_k^{-1}X^T u$ and $\tilde{y} := y^k + \tau_k^{-1}u$. By (Zhang et al., 2020, (10) and Theorem 3.1), we can construct a matrix $H \in \widehat{\partial}(\nabla\Psi)(\cdot)$ as follows

$$H := \sigma_k^{-1}\sum_{l=1}^{g}\left[X_{G_l}\boldsymbol{V}_{\frac{\lambda w_2\omega_l}{\sigma_k}}\left(\mathtt{prox}_{\frac{\lambda w_1}{\sigma_k}\|\cdot\|_1}(\tilde{\beta}_{G_l})\right)\boldsymbol{U}_{\frac{\lambda w_1}{\sigma_k}}(\tilde{\beta}_{G_l})X_{G_l}^T\right] + \tau_k^{-1}\boldsymbol{V}_{\frac{1}{\tau_k}}(\tilde{y}).$$

**Fused Lasso regularizer.** Next, we illustrate the construction of an element in $\widehat{\partial}(\nabla\Psi)(\cdot)$ when $p$ is the fused Lasso regularizer given by (7). From (Li et al., 2018b), we can construct a matrix $\boldsymbol{W}_{\kappa_2}(\beta)$ in $\partial\mathtt{prox}_{\kappa_2\|B\cdot\|_1}(\beta)$ for any $\beta \in \mathbb{R}^n$ as follows

$$\partial\mathtt{prox}_{\kappa_2\|B\cdot\|_1}(\beta) \ni \boldsymbol{W}_{\kappa_2}(\beta) := I - B^T(\boldsymbol{\Sigma}BB^T\boldsymbol{\Sigma})^\dagger B,$$

where $\boldsymbol{\Sigma} = \mathtt{Diag}(\boldsymbol{\sigma})$ and $\boldsymbol{\sigma}$ is given by

$$\boldsymbol{\sigma}_i = \begin{cases} 1, & \text{if } \left(B\mathtt{prox}_{\kappa_2\|B\cdot\|_1}(\beta)\right)_i = 0, \\ 0, & \text{otherwise,} \end{cases} \qquad i = 1, 2, \ldots, n-1.$$

Let $\tilde{\beta} := \beta^k - \sigma_k^{-1}X^T u$ and $\tilde{y} := y^k + \tau_k^{-1}u$. By (Li et al., 2018b, (22) and Theorem 2), we can construct a matrix $H \in \widehat{\partial}(\nabla\Psi)(\cdot)$ as follows

$$H := \sigma_k^{-1}X\boldsymbol{U}_{\frac{\lambda w_1}{\sigma_k}}\left(\mathtt{prox}_{\frac{\lambda w_2}{\sigma_k}\|B\cdot\|_1}(\tilde{\beta})\right)\boldsymbol{W}_{\frac{\lambda w_2}{\sigma_k}}(\tilde{\beta})X^T + \tau_k^{-1}\boldsymbol{V}_{\frac{1}{\tau_k}}(\tilde{y}).$$

## 4. Numerical Experiments

In this section, we aim to evaluate the performance of our proximal point dual semismooth Newton algorithm (PPDNA) for solving the square-root regularized problem (3), when the regularizer $p$ is chosen to be the sparse group Lasso regularizer (6) or the fused Lasso regularizer (7).

### 4.1 Setup of the Experiments

Let `tol` be the tolerance, `maxiter` be the maximum iteration number, and `maxtime` be the maximum running time. We terminate a method at the $k$th iteration if the $k$th iterative point $\beta^k$ satisfies one of the following conditions:

- $\left\|X\beta^k - Y\right\| \neq 0$, $\bar{\beta}^k := X^T \frac{X\beta^k - Y}{\left\|X\beta^k - Y\right\|}$, and

$$\Delta^k_{\rm kkt} := \frac{\left\|\beta^k - \mathtt{prox}_{\lambda p}(\beta^k - \bar{\beta}^k)\right\|}{1 + \|\beta^k\| + \left\|\bar{\beta}^k\right\|} < \mathtt{tol}; \tag{26}$$

- $\left\|X\beta^k - Y\right\| = 0$; in this case, if the primal objective value in (20) (denoted as $\mathrm{obj}_{(20)}$) and the dual objective value in (21) (denoted as $\mathrm{obj}_{(21)}$) are available, we report the relative duality gap

$$\Delta^k_{\rm pd.gap} := \frac{\mathrm{obj}_{(20)} - \mathrm{obj}_{(21)}}{1 + \left|\mathrm{obj}_{(20)}\right| + \left|\mathrm{obj}_{(21)}\right|}; \tag{27}$$

  otherwise, we report the relative successive change

$$\Delta^k_{\rm var.gap} := \frac{\left\|\beta^k - \beta^{k-1}\right\|}{1 + \|\beta^k\| + \|\beta^{k-1}\|}; \tag{28}$$

- $k >$ `maxiter`, or the total running time exceeds `maxtime`.

For all the algorithms, we set `tol` to be $10^{-7}$ and `maxtime` to be 30 minutes. In addition, we set `maxiter` for our algorithm to be $10^2$, and for other algorithms to be $10^6$. All the experiments are performed in MATLAB (version 9.7) on a Windows workstation (6-core, Intel Core i7-8750H @ 2.20GHz, 8 Gigabytes of RAM).

For the data matrix $X \in \mathbb{R}^{N\times n}$, we require each column of $X$ to be nonzero, i.e., $\sum_{i=1}^N X_{ij}^2 > 0$, $j = 1, 2, \ldots, n$. For all the data matrices in our experiments, we normalize the columns such that the diagonal entries of the matrix $\frac{1}{N}X^TX$ are equal to one. That is, we let $\boldsymbol{d} \in \mathbb{R}^n$ be defined by $\boldsymbol{d}_j = \sqrt{\frac{N}{\sum_{i=1}^N X_{ij}^2}}$, $j = 1, 2, \ldots, n$ and then normalize $X$ by $X \leftarrow X\mathtt{Diag}(\boldsymbol{d})$. Such a normalization of the data has been considered in (Bunea et al., 2013; Stucky and van de Geer, 2017; Blanchet et al., 2019a; Jiang et al., 2021). For the sparse group Lasso regularizer, we always choose the weights of groups as $\omega_l = \sqrt{|G_l|}$.

We also give the following explanations for the entries in the tables of numerical results. We report an estimation of the number of nonzero elements in a computed vector $\beta \in \mathbb{R}^n$ as follows

$$\mathrm{nnz}(\beta) := \mathop{\arg\min}_{1\leq j\leq n} \Big\{ j \,\Big|\, \sum_{i=1}^{j} \left|\beta_i^{\downarrow}\right| \geq 0.999 \left\|\beta\right\|_1 \Big\}, \tag{29}$$

where $\beta^{\downarrow}$ is a vector obtained from $\beta$ by sorting its entries in nonincreasing order of magnitude. When $\beta$ has a group structure $\{G_l\}_{l=1}^g$, which forms a partition of $\{1, 2, \ldots, n\}$, we define $b \in \mathbb{R}^g$ by $b_l := \left\|\beta_{G_l}\right\|$, $l = 1, \ldots, g$ and report an estimation of the number of

nonzero groups of $\beta$ by applying (29) to the vector $b$: $\mathrm{nnzgrp}(\beta) := \mathrm{nnz}(b)$. Besides, we denote $\mathrm{nnzB}(\beta) := \mathrm{nnz}(B\beta)$ where $B\beta = (\beta_1 - \beta_2, \beta_2 - \beta_3, \ldots, \beta_{n-1} - \beta_n)^T$. We show the number of outer PPA iterations in Algorithm 1 and the total number of inner SSN iterations in Algorithm 2 of our PPDNA in the format of "outer iteration | inner iteration" under the iteration column. The running time is in the format of "minutes:seconds". An entry "00" under the column "time" means that the computational time is less than 0.5 second. Based on the value of $\|X\beta^k - Y\|$, we report under the error column $\Delta_{\mathrm{kkt}}$, $*\Delta_{\mathrm{pd.gap}}$, or $\#\Delta_{\mathrm{var.gap}}$, given by (26), (27), and (28), respectively.

### 4.2 Alternating Direction Method of Multipliers for Solving the Square-root Regularized Problem

To justify the necessity of our second order based method PPDNA in Section 3, here we develop two first order methods based on the highly popular alternating direction method of multipliers (ADMM) (Glowinski and Marroco, 1975; Gabay and Mercier, 1976) framework to compare our PPDNA against them.

Now we describe the implementation of ADMM for solving an equivalent form of the square-root regularized problem (20) and its dual problem (21). By introducing slack variables $\alpha \in \mathbb{R}^n$, $\xi \in \mathbb{R}^n$, and $x \in \mathbb{R}^N$, we obtain their equivalent forms respectively as follows

$$\min_{\beta\in\mathbb{R}^n,\, y\in\mathbb{R}^N,\, \alpha\in\mathbb{R}^n} \left\{ \|y\| + \lambda p(\alpha) \mid X\beta - Y - y = 0, \beta - \alpha = 0 \right\}, \tag{30}$$

$$- \min_{u\in\mathbb{R}^N,\, \xi\in\mathbb{R}^n,\, x\in\mathbb{R}^N} \left\{ \langle Y, u\rangle + (\lambda p)^*(\xi) + \delta_{\mathcal{B}}(x) \mid -X^T u - \xi = 0, u - x = 0 \right\}. \tag{31}$$

Given a positive scalar $\mu$, the augmented Lagrangian functions associated with (30) and (31) are respectively given by

$$\begin{aligned}
\mathcal{L}_\mu^{(\mathrm{P})}(\beta, y, \alpha; u, \xi) &= \|y\| + \lambda p(\alpha) + \frac{\mu}{2}\left\|X\beta - Y - y + \mu^{-1}u\right\|^2 + \frac{\mu}{2}\left\|\beta - \alpha + \mu^{-1}\xi\right\|^2 \\
&\quad - \frac{1}{2\mu}\|u\|^2 - \frac{1}{2\mu}\|\xi\|^2, \quad \forall\ (\beta, y, \alpha, u, \xi) \in \mathbb{R}^n \times \mathbb{R}^N \times \mathbb{R}^n \times \mathbb{R}^N \times \mathbb{R}^n, \\
\mathcal{L}_\mu^{(\mathrm{D})}(u, \xi, x; \beta, y) &= \langle Y, u\rangle + \delta_{\mathcal{B}}(x) + (\lambda p)^*(\xi) + \frac{\mu}{2}\left\|-X^T u - \xi + \mu^{-1}\beta\right\|^2 + \frac{\mu}{2}\left\|u - x + \mu^{-1}y\right\|^2 \\
&\quad - \frac{1}{2\mu}\|\beta\|^2 - \frac{1}{2\mu}\|y\|^2, \quad \forall (u, \xi, x, \beta, y) \in \mathbb{R}^N \times \mathbb{R}^n \times \mathbb{R}^N \times \mathbb{R}^n \times \mathbb{R}^N.
\end{aligned}$$

The ADMM for solving the primal problem (30) and the dual problem (31) is given respectively in Algorithm 3 and Algorithm 4. For the convergence results, we refer the reader to (Chen et al., 2017; Fazel et al., 2013). We make the following remarks on techniques for solving linear systems (32) and (33), which is the most expensive part in ADMM frameworks. When $n < N$, we solve the $n \times n$ linear system (32) either by the Cholesky factorization or the preconditioned conjugate gradients method, depending on $n$. Otherwise, we apply the Sherman-Morrison-Woodbury formula (Golub and Van Loan, 2013)

$$(I + X^T X)^{-1} = I - X(I + XX^T)^{-1}X^T,$$

and only compute the Cholesky factorization of a smaller $N \times N$ matrix $I + XX^T$. For solving the $N \times N$ linear system (33) when $N < n$, we use either the Cholesky factorization

or the preconditioned conjugate gradients method, depending on $N$. Otherwise, we apply the Sherman-Morrison-Woodbury formula

$$(I + XX^T)^{-1} = I - X^T(I + X^TX)^{-1}X,$$

and compute the Cholesky factorization of $I+X^TX$. In any case, the Cholesky factorization (if needed) is only performed once at the beginning of the algorithm.

**Algorithm 3:** pADMM for solving (30)

**Input** Data $X \in \mathbb{R}^{N\times n}$, $Y \in \mathbb{R}^N$, a penalty parameter $\lambda > 0$, a regularizer $p$.
**Initialize**
$k = 0, y^0 = \mathbf{0} \in \mathbb{R}^N, \alpha^0 = \mathbf{0} \in \mathbb{R}^n, u^0 = \mathbf{0} \in \mathbb{R}^N, \xi^0 = \mathbf{0} \in \mathbb{R}^n; \rho = 1.618, \mu > 0.$
**while** *the termination criterion is not met,* **do**

**Step 1.** Compute $\beta^{k+1}$ by

$$\begin{aligned}\beta^{k+1} &= \arg\min_{\beta} \frac{\mu}{2}\left\|X\beta - Y - y^k + \mu^{-1}u^k\right\|^2 + \frac{\mu}{2}\left\|\beta - \alpha^k + \mu^{-1}\xi^k\right\|^2 \\ &= (I + X^TX)^{-1}\left[X^T(Y + y^k - \mu^{-1}u^k) + (\alpha^k - \mu^{-1}\xi^k)\right]. \end{aligned} \tag{32}$$

**Step 2.** Compute $y^{k+1}, \alpha^{k+1}$ by

$$\begin{aligned} y^{k+1} &= \arg\min_{y} \|y\| + \frac{\mu}{2}\left\|X\beta^{k+1} - Y - y + \mu^{-1}u^k\right\|^2 \\ &= \mathtt{prox}_{\mu^{-1}\|\cdot\|}\left(X\beta^{k+1} - Y + \mu^{-1}u^k\right), \\ \alpha^{k+1} &= \arg\min_{\alpha} \lambda p(\alpha) + \frac{\mu}{2}\left\|\beta^{k+1} - \alpha + \mu^{-1}\xi^k\right\|^2 = \mathtt{prox}_{\mu^{-1}\lambda p}\left(\beta^{k+1} + \mu^{-1}\xi^k\right). \end{aligned}$$

**Step 3.** Update $u^{k+1}, \xi^{k+1}$ by

$$u^{k+1} = u^k + \rho\mu(X\beta^{k+1} - Y - y^{k+1}), \qquad \xi^{k+1} = \xi^k + \rho\mu(\beta^{k+1} - \alpha^{k+1}).$$

**Step 4.** Set $k \leftarrow k + 1$.
**end**
**Output** $\beta^k$.

### 4.3 Comparison of Efficiency for Solving the Square-root Sparse Group Lasso Problem

In this section, we conduct extensive experiments to demonstrate the efficiency of our PPDNA for solving the square-root sparse group Lasso problem where the regularizer $p$ is given by (6). In particular, we compare our PPDNA (Algorithm 1+Algorithm 2) with pADMM (Algorithm 3), dADMM (Algorithm 4), and the S-TISP solver (Bunea et al., 2013). Note that the S-TISP solver is limited to solving the square-root group Lasso problem where the coefficients in the expression (6) of the regularier $p$ can only be taken as

**Algorithm 4:** dADMM for solving (31)

**Input** Data $X \in \mathbb{R}^{N\times n}$, $Y \in \mathbb{R}^N$, a penalty parameter $\lambda > 0$, a regularizer $p$.
**Initialize** $k = 0, u^0 = \mathbf{0} \in \mathbb{R}^N, \xi^0 = \mathbf{0} \in \mathbb{R}^n, x^0 = \mathbf{0} \in \mathbb{R}^N, \beta^0 = \mathbf{0} \in \mathbb{R}^n, y^0 = \mathbf{0} \in \mathbb{R}^N; \rho = 1.618, \mu > 0$.
**while** *the termination criterion is not met,* **do**

**Step 1.** Compute $u^{k+1}$ by

$$\begin{aligned} u^{k+1} &= \arg\min_u u^T Y + \frac{\mu}{2}\left\|-X^T u - \xi^k + \mu^{-1}\beta^k\right\|^2 + \frac{\mu}{2}\left\|u - x^k + \mu^{-1}y^k\right\|^2 \\ &= (I + XX^T)^{-1}\left[-\mu^{-1}Y + X(\mu^{-1}\beta^k - \xi^k) - (\mu^{-1}y^k - x^k)\right]. \end{aligned} \tag{33}$$

**Step 2.** Compute $\xi^{k+1}, x^{k+1}$ by

$$\begin{aligned} \xi^{k+1} &= \arg\min_\xi (\lambda p)^*(\xi) + \frac{\mu}{2}\left\|-X^T u^{k+1} - \xi + \mu^{-1}\beta^k\right\|^2 \\ &= \mathtt{prox}_{\mu^{-1}(\lambda p)^*}\left(-X^T u^{k+1} + \mu^{-1}\beta^k\right) \\ &= \left(-X^T u^{k+1} + \mu^{-1}\beta^k\right) - \mu^{-1}\mathtt{prox}_{\mu\lambda p}\left(-\mu X^T u^{k+1} + \beta^k\right), \\ x^{k+1} &= \arg\min_x \delta_{\mathcal{B}}(x) + \frac{\mu}{2}\left\|u^{k+1} - x + \mu^{-1}y^k\right\|^2 = \Pi_{\mathcal{B}}\left(u^{k+1} + \mu^{-1}y^k\right). \end{aligned}$$

**Step 3.** Update $\beta^{k+1}, y^{k+1}$ by

$$\beta^{k+1} = \beta^k + \rho\mu(-X^T u^{k+1} - \xi^{k+1}), \qquad y^{k+1} = y^k + \rho\mu(u^{k+1} - x^{k+1}).$$

**Step 4.** Set $k \leftarrow k + 1$.

**end**
**Output** $\beta^k$.

$(w_1, w_2) = (0, 1)$ or $(w_1, w_2) = (1, 0)$ in that solver. For all tables in this section, we denote PPDNA, pADMM, dADMM, and S-TISP by "PP", "pA", "dA", and "ST", respectively.

#### 4.3.1 Synthetic Data

We first show the results on synthetic data sets following the data generation mechanism in (Yuan and Lin, 2006; Bunea et al., 2013; Blanchet and Kang, 2017). We first choose a correlation matrix $\Sigma$ to be a Toeplitz matrix, i.e., $\Sigma_{ij} = 0.5^{|i-j|}$, the dimensions of which will be clear from the context. We construct four examples in our experiments. As can be seen from the following examples, the true regression vector $\beta_0$ designed in Example 3 is not only groupwise sparse, but also sparse within a group. In contrast, the nonzero groups for $\beta_0$ in Example 1 and 2 are dense. Besides, based on Example 3, we construct Example 4a/4b for which the true regression vectors are denser.

**Example 1** (Bunea et al., 2013) The dimensions of this example are set as $N = 1000$, $g = 200$ or $2000$, $n = 3g$. We assign every three adjacent entries to be in one group,

i.e., $G_l = \{3l-2, 3l-1, 3l\}$, $l = 1, \ldots, g$. The true regression vector $\beta_0$ only contains three nonzero groups, i.e., $(\beta_0)_{G_l} = (2.5, 2.5, 2.5)^T$ for $l = 1, 3, 4$ and $(\beta_0)_{G_l} = \mathbf{0}$ otherwise. The predictor vectors $X_i \in \mathbb{R}^n$, $i = 1, \ldots, N$ are generated from the multivariate normal distribution $\mathcal{N}(\mathbf{0}, \Sigma)$. The response variables $Y_i \in \mathbb{R}$, $i = 1, \ldots, N$ are generated from the linear regression model (1) with $\epsilon_i \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, 1)$ and $\sigma = 1$.

**Example 2** (Yuan and Lin, 2006; Blanchet and Kang, 2017) The dimensions of this example are set as $N = 500$ or $10000$, $g = 160$, $n = 3g$. The group structure is given by $G_l = \{l, l+g, l+2g\}$, $l = 1, \ldots, g$. The true regression vector $\beta_0$ only contains two nonzero groups, i.e., $(\beta_0)_{G_3} = (1, 1, 1)^T$, $(\beta_0)_{G_6} = (2/3, -1, 1/2)^T$, and $(\beta_0)_{G_l} = \mathbf{0}$, $l \notin \{3, 6\}$. We generate random vectors $Z_i \in \mathbb{R}^g$, $i = 1, \ldots, N$ from $\mathcal{N}(\mathbf{0}, \Sigma)$ and a random scalar $\omega \in \mathbb{R}$ from $\mathcal{N}(0, 1)$. Let $A_i := \frac{Z_i + \omega \mathbf{1}}{\sqrt{2}} \in \mathbb{R}^g$. Then the predictor vectors $X_i \in \mathbb{R}^n$, $i = 1, \ldots, N$ are chosen to be the concatenation of three vectors constructed from $A_i$: $X_i := (A_i^T, A_i^T \odot A_i^T, A_i^T \odot A_i^T \odot A_i^T)^T$. The response variables $Y_i \in \mathbb{R}$, $i = 1, \ldots, N$ are generated from the linear regression model (1) with $\epsilon_i \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, 1)$ and $\sigma = 2$.

**Example 3** The dimensions of this example are set as $N = 4000$, $g = 2000$, $n = 3g$ or $N = 500$, $g = 3000$, $n = 3g$. The true regression vector $\beta_0$ only contains four nonzero groups, i.e., $(\beta_0)_{G_3} = (1, 0, 1)^T$, $(\beta_0)_{G_6} = (2/3, -1, 0)^T$, $(\beta_0)_{G_9} = (-1, 0, -1/2)^T$, $(\beta_0)_{G_{12}} = (0, -1, 0)^T$, and $(\beta_0)_{G_l} = \mathbf{0}$, $l \notin \{3, 6, 9, 12\}$. The group structure $\{G_l\}$ and the data $\{X, Y\}$ are generated in the same way as in **Example 2**.

**Example 4a/4b** The dimensions of this example are set as $N = 4000, g = 2000, n = 3g$. Let $\beta_0$ be the true regression vector in **Example 3**. For a positive integer $T$, we define $\beta^{(T)} \in \mathbb{R}^n$ as follows:

$$\beta^{(T)}_{G_l} = \begin{cases} (\beta_0)_{G_i} & \text{if } l = 12k + i,\ k \in \{0, 1, \ldots, T-1\},\ i \in \{1, 2, \ldots, 12\}, \\ \mathbf{0} & \text{otherwise.} \end{cases}$$

By this construction, we have $\text{nnz}\left(\beta^{(T)}\right) = 7T$ and $\text{nnzgrp}\left(\beta^{(T)}\right) = 4T$. In particular, we set $\beta^{(10)}$ and $\beta^{(100)}$ as the true regression vectors of **Example 4a** and **Example 4b**, respectively. The group structure $\{G_l\}$ and the data set $\{X, Y\}$ are generated in the same way as in **Example 3**.

We choose $\lambda \in \{\lambda_{\text{Bun}}, \lambda_{\text{StG}}, \lambda_{\text{BlG}}\}$, where the three tuning parameters $\lambda_{\text{Bun}}$, $\lambda_{\text{StG}}$, and $\lambda_{\text{BlG}}$ are theoretically optimal values given in (Bunea et al., 2013; Stucky and van de Geer, 2017; Blanchet et al., 2019a). See Appendix B for the details of the choice of the parameters. First, we set $w_1 = 0$ and $w_2 = 1$ so that we can compare with the S-TISP solver, and the comparison of PPDNA, pADMM, dADMM, and S-TISP on Examples 1-4 are reported in Table 1. We can see from Table 1 that our PPDNA successfully solves all instances except for Example 4b within several seconds. For Example 3 with $(N, n) = (4000, 6000)$, pADMM fails to achieve the prespecified accuracy within 30 minutes, dADMM takes approximately 20 minutes, and the running time of S-TISP varies from about 2 minutes to 12 minutes; however, our PPDNA merely takes 5 seconds. In addition, we set $w_1 = w_2 = 0.5$ and compare the three methods PPDNA, pADMM, and dADMM in Table 2. It can be observed that our PPDNA outperforms both pADMM and dADMM for all instances in Table 2; our PPDNA returns accurate solutions for all instances within one minute. Even though dADMM is faster than pADMM, its running time grows significantly with the increase of dimensions. In addition, comparing Examples 3, 4a, 4b ($N = 4000, n = 6000, g = 2000$)

where the numbers of nonzeros of the true regression vectors are 7, 70, 700, respectively, we found that the performances of the algorithms do not vary too much. By comparing Table 1 and Table 2, one may find that the efficiency of our PPDNA remains steady for the group Lasso regularizer ($w_1 = 0$) and the sparse group Lasso regularizer ($w_1 = 0.5$).

#### 4.3.2 UCI Data with Synthetic Group Structure

We use the UCI repository (Asuncion and Newman, 2007; Chang and Lin, 2011) in this section. Following (Huang et al., 2010; Li et al., 2018c; Zhang et al., 2020), we expand the original features of the data sets *housing*, *bodyfat*, *pyrim*, *triazines*. The UCI data sets are not equipped with group structures, and we design the group structure as follows. We set the total number of groups as $g = 300$, and a feature is assigned randomly to a group with a uniform probability.

On each UCI data set, we consider two cases: $(w_1, w_2) = (0, 1)$ and $(w_1, w_2) = (0.5, 0.5)$. For each case, we select three values of $\lambda$ from the set $\{0.5, 1, 2, 5, 7.5\}$ so that the resulting solutions do not overfit. When $w_1 = 0$, we compare the performances of PPDNA with pADMM, dADMM, and S-TISP. Otherwise, we compare the performances of PPDNA with pADMM and dADMM. The results are presented in Table 3. It can be seen that PPDNA is always the fastest among the compared methods for all instances in Table 3. We also find that the iteration numbers and computational time of pADMM, dADMM, and S-TISP fluctuates wildly with the changes in the dimensions or parameters. In contrast, the performances of our PPDNA are generally robust; it solves all the problems in Table 3 within 30 seconds except for *E2006.train*. In addition, we can observe that pADMM is generally slow for problems in Table 3, and it can not solve the last problem due to insufficient of memory for computing the Cholesky factorization. Compared to S-TISP, dADMM is faster for the first four problems. For the last two problems, dADMM fails to solve them within 30 minutes, which is due to the computational cost for solving the linear system (33). But note that S-TISP does not solve some of instances to the required level of accuracy. Since the dimensions of problems in UCI data sets are much larger than those of the synthetic data sets in Section 4.3.1, pADMM, dADMM, and S-TISP reach the maximum running time 30 minutes in many instances. In contrast, our PPDNA succeeds in solving all instances. We can safely conclude that our PPDNA can be more efficient than pADMM, dADMM, and S-TISP for solving large-scale square-root sparse group Lasso problems.

#### 4.3.3 Real Data

In this section, we present the numerical results of the square-root sparse group Lasso model on two real data sets which are equipped with natural group structures. For a given data set $(X, Y)$ in this section, we randomly split it into the training set $(X_{\text{train}}, Y_{\text{train}})$ and the test set $(X_{\text{test}}, Y_{\text{test}})$ so that the number of observations $N_{\text{train}}$ of the training data set is roughly twice larger than the number of observations $N_{\text{test}}$ of the test set. Based on the training set, we first set $(w_1, w_2) = (0, 1)$ and conduct 8-fold cross validation (CV) for selecting $\lambda$ over the set

$$\lambda \in \{10^{-1}, 10^{-0.95}, 10^{-0.9}, \ldots, 10^{0.95}, 10^{1}\}. \tag{34}$$

| problem | $\lambda$ | nnz\| | iter | | | | time | | | | error | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $(N,n)$; $g$ | | nnzgrp | PP | pA | dA | ST | PP | pA | dA | ST | PP | pA | dA | ST |
| Example 1 | 3.485 | 9\|3 | 17\|70 | 1489 | 896 | 46 | 01 | 02 | 01 | 00 | 3.4e-08 | 4.6e-08 | 4.5e-08 | 8.8e-08 |
| (1000, 600) | 9.193 | 9\|3 | 13\|41 | 1071 | 1291 | 123 | 00 | 01 | 02 | 00 | 7.7e-09 | 5.3e-08 | 1.2e-08 | 9.4e-08 |
| 200 | 4.929 | 9\|3 | 15\|66 | 1489 | 1079 | 43 | 01 | 02 | 02 | 00 | 4.4e-09 | 3.3e-08 | 1.9e-08 | 7.8e-08 |
| Example 1 | 3.929 | 9\|3 | 13\|28 | 14374 | 6235 | 84 | 04 | 03:27 | 56 | 00 | 3.0e-11 | 5.3e-08 | 2.1e-08 | 8.8e-08 |
| (1000, 6000) | 9.837 | 9\|3 | 11\|28 | 11054 | 3211 | 63 | 04 | 02:45 | 30 | 00 | 2.0e-08 | 2.0e-08 | 3.9e-08 | 8.5e-08 |
| 2000 | 5.605 | 9\|3 | 11\|24 | 27501 | 6058 | 77 | 04 | 06:42 | 54 | 00 | 5.2e-08 | 3.7e-08 | 3.3e-08 | 8.7e-08 |
| Example 2 | 3.462 | 6\|2 | 14\|48 | 12501 | 2993 | 2178 | 00 | 09 | 02 | 00 | 4.1e-09 | 1.6e-08 | 9.9e-08 | 1.0e-07 |
| (500, 480) | 9.193 | 6\|2 | 13\|43 | 6389 | 3041 | 529 | 00 | 04 | 02 | 00 | 6.1e-08 | 2.4e-08 | 1.9e-08 | 9.9e-08 |
| 160 | 4.601 | 6\|2 | 14\|45 | 10001 | 3421 | 1860 | 00 | 07 | 02 | 00 | 1.1e-09 | 1.8e-08 | 2.5e-08 | 1.0e-07 |
| Example 2 | 3.353 | 6\|2 | 14\|53 | 100001 | 9442 | 3527 | 08 | 09:49 | 01:59 | 14 | 4.5e-09 | 7.0e-08 | 1.0e-07 | 1.0e-07 |
| (10000, 480) | 9.020 | 6\|2 | 14\|52 | 60001 | 16444 | 2574 | 08 | 05:54 | 03:28 | 10 | 1.6e-09 | 9.9e-08 | 9.9e-08 | 1.0e-07 |
| 160 | 4.546 | 6\|2 | 13\|55 | 72501 | 10014 | 3053 | 08 | 07:09 | 02:12 | 12 | 3.6e-08 | 8.6e-08 | 9.9e-08 | 1.0e-07 |
| Example 3 | 4.026 | 23\|8 | 13\|37 | 27501 | 9581 | 27607 | 05 | 04:50 | 01:00 | 02:14 | 2.8e-09 | 7.0e-08 | 3.7e-08 | 1.0e-07 |
| (500, 9000) | 10.008 | 3\|1 | 12\|28 | 95001 | 15001 | 5870 | 05 | 17:25 | 01:37 | 28 | 1.2e-09 | 1.7e-08 | 3.0e-08 | 1.0e-07 |
| 3000 | 17.169 | 0\|0 | 1\|1 | 1 | 1 | 1 | 00 | 00 | 00 | 00 | 0.0e-00 | 0.0e-00 | 0.0e-00 | 0.0e-00 |
| Example 3 | 3.800 | 12\|4 | 14\|51 | 24123 | 19459 | 34664 | 05 | 30:00 | 16:50 | 12:51 | 1.7e-09 | 7.1e-05 | 7.4e-08 | 1.0e-07 |
| (4000, 6000) | 9.760 | 12\|4 | 14\|48 | 23646 | 22501 | 21300 | 05 | 30:00 | 19:52 | 08:07 | 3.6e-11 | 4.8e-06 | 1.8e-08 | 1.0e-07 |
| 2000 | 17.163 | 12\|4 | 13\|48 | 23978 | 25001 | 7288 | 05 | 30:00 | 21:48 | 02:54 | 7.6e-08 | 7.7e-06 | 6.5e-08 | 1.0e-07 |
| Example 4a | 4.005 | 111\|41 | 21\|118 | 34205 | 7409 | 36603 | 06 | 30:00 | 05:21 | 06:45 | 4.1e-09 | 1.8e-07 | 1.0e-07 | 1.0e-07 |
| (4000, 6000) | 9.791 | 132\|48 | 21\|111 | 10001 | 19713 | 19748 | 03 | 08:49 | 13:57 | 03:35 | 1.3e-08 | 2.1e-08 | 3.2e-08 | 1.0e-07 |
| 2000 | 16.910 | 110\|37 | 20\|90 | 22501 | 25001 | 18372 | 02 | 19:51 | 17:40 | 03:20 | 6.3e-09 | 4.1e-08 | 4.4e-08 | 1.0e-07 |
| Example 4b | 3.994 | 1256\|447 | 18\|103 | 33392 | 4233 | 29186 | 01:33 | 30:00 | 03:11 | 05:22 | 3.8e-09 | 4.9e-07 | 9.8e-08 | 1.0e-07 |
| (4000, 6000) | 9.791 | 1142\|387 | 21\|127 | 10001 | 7271 | 29179 | 26 | 09:04 | 05:22 | 05:21 | 1.4e-08 | 7.5e-08 | 1.0e-07 | 1.0e-07 |
| 2000 | 17.466 | 852\|290 | 22\|143 | 12501 | 13635 | 22256 | 20 | 11:13 | 09:51 | 04:04 | 1.6e-08 | 2.7e-08 | 2.3e-08 | 1.0e-07 |

Table 1: Square-root sparse group Lasso model on synthetic data sets Examples 1-4 with $(w_1, w_2) = (0, 1)$.

| problem | $\lambda$ | nnz\| | iter | | | time | | | error | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $(N,n)$; $g$ | | nnzgrp | PP | pA | dA | PP | pA | dA | PP | pA | dA |
| Example 1 | 3.589 | 9\|3 | 14\|58 | 1489 | 896 | 01 | 02 | 02 | 1.6e-08 | 4.5e-08 | 4.5e-08 |
| (1000, 600) | 9.262 | 9\|3 | 13\|44 | 1071 | 1291 | 00 | 01 | 02 | 1.5e-08 | 4.9e-08 | 1.1e-08 |
| 200 | 5.018 | 9\|3 | 18\|79 | 1489 | 1081 | 01 | 02 | 02 | 2.4e-09 | 3.2e-08 | 1.8e-08 |
| Example 1 | 4.064 | 9\|3 | 12\|26 | 22501 | 6243 | 04 | 05:46 | 01:00 | 2.7e-10 | 8.2e-08 | 1.5e-08 |
| (1000, 6000) | 9.906 | 9\|3 | 11\|28 | 9581 | 3211 | 04 | 02:28 | 31 | 4.2e-08 | 3.7e-08 | 8.9e-08 |
| 2000 | 5.615 | 9\|3 | 11\|24 | 15001 | 6210 | 04 | 03:51 | 59 | 7.5e-08 | 2.6e-08 | 4.9e-09 |
| Example 2 | 3.724 | 6\|3 | 14\|56 | 10001 | 2892 | 01 | 08 | 02 | 5.6e-09 | 1.2e-08 | 9.9e-08 |
| (500, 480) | 9.298 | 6\|2 | 14\|51 | 5325 | 3062 | 01 | 04 | 02 | 3.2e-08 | 9.3e-08 | 1.2e-08 |
| 160 | 4.593 | 6\|2 | 14\|53 | 10001 | 3286 | 01 | 08 | 02 | 1.7e-09 | 1.4e-08 | 9.0e-08 |
| Example 2 | 3.454 | 6\|3 | 18\|81 | 52501 | 8454 | 10 | 05:29 | 01:53 | 1.9e-09 | 6.9e-08 | 1.0e-07 |
| (10000, 480) | 9.039 | 6\|2 | 15\|60 | 60001 | 13449 | 08 | 06:15 | 02:57 | 5.0e-10 | 9.4e-08 | 4.2e-08 |
| 160 | 4.564 | 6\|2 | 16\|79 | 72501 | 14916 | 11 | 07:33 | 03:22 | 6.1e-09 | 8.6e-08 | 9.9e-08 |
| Example 3 | 4.422 | 22\|9 | 13\|34 | 32501 | 7986 | 05 | 05:49 | 54 | 2.1e-11 | 6.1e-08 | 6.3e-08 |
| (500, 9000) | 10.114 | 3\|1 | 12\|29 | 35001 | 11496 | 05 | 06:21 | 01:15 | 7.9e-10 | 6.2e-08 | 8.3e-08 |
| 3000 | 17.216 | 0\|0 | 1\|1 | 1 | 1 | 00 | 00 | 00 | 0.0e-00 | 0.0e-00 | 0.0e-00 |
| Example 3 | 3.955 | 10\|5 | 14\|50 | 23062 | 22896 | 05 | 30:00 | 20:45 | 4.2e-10 | 8.0e-05 | 1.0e-07 |
| (4000, 6000) | 9.791 | 10\|4 | 14\|54 | 22845 | 22046 | 05 | 30:00 | 20:05 | 1.3e-09 | 6.0e-06 | 3.6e-08 |
| 2000 | 16.958 | 8\|4 | 12\|39 | 22889 | 22501 | 05 | 30:00 | 20:26 | 7.8e-08 | 2.7e-06 | 8.2e-08 |
| Example 4a | 4.005 | 115\|72 | 20\|106 | 34158 | 11104 | 03 | 30:00 | 08:09 | 3.5e-08 | 1.8e-06 | 1.0e-07 |
| (4000, 6000) | 9.791 | 101\|78 | 21\|104 | 10001 | 19907 | 03 | 08:49 | 14:08 | 4.3e-08 | 5.6e-08 | 6.1e-08 |
| 2000 | 16.910 | 133\|61 | 21\|98 | 15001 | 17501 | 03 | 13:14 | 12:23 | 1.5e-08 | 3.6e-08 | 4.2e-08 |
| Example 4b | 3.994 | 889\|483 | 22\|111 | 27501 | 4324 | 41 | 24:51 | 03:18 | 1.4e-08 | 8.2e-08 | 1.0e-07 |
| (4000, 6000) | 9.791 | 1114\|403 | 21\|100 | 25001 | 5183 | 21 | 22:35 | 03:51 | 5.6e-09 | 9.2e-08 | 1.0e-07 |
| 2000 | 17.466 | 884\|298 | 18\|130 | 7501 | 8190 | 19 | 06:45 | 05:56 | 8.6e-09 | 3.3e-08 | 6.9e-08 |

Table 2: Square-root sparse group Lasso model on synthetic data sets Examples 1-4 with $w_1 = w_2 = 0.5$.

| problem | $w_1\|\lambda$ | nnz\| | iter | | | | time | | | | error | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $(N,n);g$ | | nnzgrp | PP | pA | dA | ST | PP | pA | dA | ST | PP | pA | dA | ST |
| *housing* $(253, 77520)$ 300 | 0.0\|0.5 | 1846\|8 | 12\|32 | 3851 | 2470 | 86386 | 08 | 02:36 | 54 | 30:00 | 1.2e-08 | 3.8e-08 | 1.0e-07 | 1.7e-04 |
| | 0.0\|1.0 | 1640\|7 | 17\|66 | 7986 | 4341 | 88088 | 02 | 05:15 | 01:32 | 30:00 | 1.5e-08 | 4.1e-08 | 2.0e-08 | 8.8e-05 |
| | 0.0\|2.0 | 724\|3 | 16\|66 | 17501 | 7986 | 89932 | 03 | 11:26 | 02:45 | 30:00 | 1.9e-08 | 4.0e-08 | 4.2e-08 | 3.3e-05 |
| | 0.5\|0.5 | 1145\|11 | 15\|39 | 5546 | 2909 | – | 09 | 03:43 | 01:06 | – | 1.5e-10 | 2.6e-08 | 1.0e-07 | – |
| | 0.5\|1.0 | 709\|7 | 15\|61 | 11496 | 4495 | – | 02 | 07:39 | 01:37 | – | 3.5e-09 | 5.5e-08 | 3.2e-08 | – |
| | 0.5\|2.0 | 460\|4 | 13\|48 | 20001 | 9020 | – | 01 | 12:57 | 03:07 | – | 7.6e-09 | 7.5e-08 | 2.0e-08 | – |
| *bodyfat* $(126, 116280)$ 300 | 0.0\|0.5 | 703\|2 | 12\|25 | 11496 | 3081 | 104870 | 08 | 05:52 | 50 | 30:00 | 6.7e-11 | 9.9e-08 | 2.6e-08 | 6.2e-03 |
| | 0.0\|1.0 | 349\|1 | 12\|53 | 45001 | 17501 | 106770 | 01 | 22:54 | 04:42 | 30:00 | 1.4e-08 | 8.8e-08 | 5.4e-08 | 4.3e-03 |
| | 0.0\|2.0 | 348\|1 | 12\|39 | 60191 | 32501 | 107478 | 01 | 30:00 | 08:35 | 30:00 | 1.8e-11 | 2.2e-06 | 8.9e-08 | 3.3e-03 |
| | 0.5\|0.5 | 84\|3 | 14\|38 | 47501 | 10772 | – | 08 | 25:02 | 03:21 | – | 9.0e-11 | 9.7e-08 | 1.0e-07 | – |
| | 0.5\|1.0 | 85\|2 | 13\|60 | 56682 | 8317 | – | 02 | 30:00 | 02:24 | – | 5.9e-08 | 8.3e-07 | 8.7e-08 | – |
| | 0.5\|2.0 | 227\|2 | 13\|50 | 59609 | 27501 | – | 01 | 30:00 | 07:20 | – | 7.3e-09 | 1.0e-05 | 6.4e-08 | – |
| *pyrim* $(37, 169911)$ 300 | 0.0\|0.5 | 1072\|2 | 13\|27 | 45001 | 3211 | 129108 | 07 | 18:12 | 44 | 30:00 | 1.6e-09 | 6.5e-08 | 3.9e-08 | 1.3e-02 |
| | 0.0\|1.0 | 577\|1 | 11\|41 | 52501 | 7986 | 62705 | 01 | 21:16 | 01:46 | 14:05 | 3.1e-08 | 7.9e-08 | 1.1e-08 | 1.0e-07 |
| | 0.0\|2.0 | 1099\|2 | 12\|39 | 75001 | 13795 | 135011 | 01 | 29:58 | 03:03 | 30:00 | 1.9e-09 | 9.8e-08 | 1.6e-08 | 3.3e-03 |
| | 0.5\|0.5 | 653\|3 | 11\|21 | 37501 | 2300 | – | 08 | 15:24 | 32 | – | 3.1e-10 | 3.7e-08 | 5.6e-08 | – |
| | 0.5\|1.0 | 215\|1 | 12\|48 | 32501 | 12501 | – | 01 | 13:17 | 02:51 | – | 9.0e-11 | 6.3e-08 | 8.6e-08 | – |
| | 0.5\|2.0 | 350\|1 | 11\|38 | 67501 | 9581 | – | 01 | 26:40 | 02:07 | – | 8.1e-09 | 9.4e-08 | 8.2e-08 | – |
| *triazines* $(93, 557845)$ 300 | 0.0\|0.5 | 7040\|4 | 18\|75 | 7961 | 6656 | 11474 | 20 | 30:00 | 12:39 | 30:00 | 9.8e-09 | 9.9e-06 | 5.2e-08 | 5.4e-02 |
| | 0.0\|1.0 | 3556\|2 | 18\|80 | 8000 | 14856 | 12325 | 15 | 30:00 | 30:00 | 30:00 | 7.2e-09 | 9.9e-06 | 1.1e-07 | 6.3e-02 |
| | 0.0\|2.0 | 1806\|1 | 13\|57 | 8981 | 17492 | 13194 | 10 | 30:00 | 30:00 | 30:00 | 1.1e-08 | 8.9e-06 | 1.7e-07 | 1.9e-02 |
| | 0.5\|0.5 | 2809\|3 | 21\|99 | 8630 | 5546 | – | 20 | 30:00 | 09:56 | – | 8.9e-08 | 3.7e-05 | 9.2e-08 | – |
| | 0.5\|1.0 | 1987\|2 | 21\|94 | 8665 | 7986 | – | 16 | 30:00 | 14:25 | – | 2.9e-08 | 9.2e-06 | 3.0e-08 | – |
| | 0.5\|2.0 | 2959\|3 | 14\|58 | 9016 | 17286 | – | 10 | 30:00 | 30:00 | – | 2.4e-09 | 8.8e-06 | 9.1e-08 | – |
| *E2006.test* $(1654, 72812)$ 300 | 0.0\|2.0 | 5972\|37 | 15\|69 | 37501 | 12501 | 17749 | 03 | 13:23 | 03:18 | 02:06 | 3.6e-08 | 6.0e-09 | 5.4e-08 | 9.7e-08 |
| | 0.0\|5.0 | 1395\|8 | 14\|46 | 11497 | 15001 | 4100 | 00 | 03:58 | 03:42 | 27 | 8.8e-11 | 7.6e-08 | 7.3e-08 | 9.9e-08 |
| | 0.0\|7.5 | 0\|0 | 1\|1 | 1 | 1 | 1 | 00 | 00 | 00 | 00 | 0.0e-00 | 0.0e-00 | 0.0e-00 | 0.0e-00 |
| | 0.5\|2.0 | 4641\|65 | 19\|82 | 32501 | 18610 | – | 05 | 12:01 | 05:01 | – | 5.6e-09 | 6.6e-08 | 4.6e-08 | – |
| | 0.5\|5.0 | 656\|22 | 13\|49 | 35001 | 20692 | – | 03 | 12:17 | 05:25 | – | 1.3e-10 | 8.9e-08 | 1.1e-08 | – |
| | 0.5\|7.5 | 149\|6 | 12\|42 | 15001 | 15001 | – | 00 | 05:13 | 03:45 | – | 6.5e-08 | 7.4e-08 | 8.1e-08 | – |
| *E2006.train* $(8044, 150348)$ 300 | 0.0\|2.0 | 16098\|46 | 19\|94 | – | 3031 | 16517 | 05:31 | – | 30:00 | 07:44 | 9.3e-10 | – | 2.7e-01 | 9.6e-08 |
| | 0.0\|5.0 | 3088\|9 | 17\|71 | – | 3010 | 11208 | 05 | – | 30:00 | 05:03 | 2.2e-08 | – | 4.6e-04 | 9.8e-08 |
| | 0.0\|7.5 | 1092\|3 | 14\|57 | – | 3012 | 3967 | 02 | – | 30:00 | 01:47 | 1.8e-09 | – | 3.8e-04 | 1.0e-07 |
| | 0.5\|2.0 | 10427\|73 | 20\|100 | – | 3015 | – | 33 | – | 30:00 | – | 8.3e-08 | – | 3.0e-01 | – |
| | 0.5\|5.0 | 1070\|23 | 17\|71 | – | 3006 | – | 08 | – | 30:00 | – | 1.3e-08 | – | 6.2e-04 | – |
| | 0.5\|7.5 | 296\|9 | 15\|60 | – | 2865 | – | 04 | – | 30:00 | – | 9.0e-09 | – | 2.1e-02 | – |

Table 3: Square-root sparse group Lasso model on UCI data sets with $(w_1, w_2) = (0, 1)$ or $(0.5, 0.5)$. "–" denotes that the method is not applicable for the instance.

We then conduct 8-fold CV for selecting parameters $w_1$, $w_2$, and $\lambda$ over the sets

$$w_1 \in \{0, 0.1, 0.2, \ldots, 0.9, 1\},\ w_2 = 1 - w_1,\ \text{and}\ \lambda \in \{10^{-1}, 10^{-0.95}, 10^{-0.9}, \ldots, 10^{0.95}, 10^{1}\}. \tag{35}$$

Based on the test set, we report the mean squared error (MSE) for an approximate solution $\beta$ defined by $\|X_{\text{test}}\beta - Y_{\text{test}}\|^2/N_{\text{test}}$.

**Climate data** (Kalnay et al., 1996) The data records climate information of 10512 locations across the globe ($73 \times 144$ grid of latitude and longitude, resolution $2.5^0 \times 2.5^0$). For each location, it records the monthly means of 7 predictor variables *Air Temperature*, *Precipitable water*, *Relative humidity*, *Pressure*, *Sea Level Pressure*, *Horizontal Wind Speed*, and *Vertical Wind Speed*. The predictor vector $X_i \in \mathbb{R}^{73584}$ is the concatenation of the 7 predictor

variables at 10512 locations in the $i$th month, and we have data from 1948/1/1 to 2018/5/31 containing $N = 814$ months. We regard the 7 predictor variables at a location as a group. The response variable is designed as follows. We first select a target location and then set $Y_i$ to be the *Air Temperature* at the target location in the $i$th month. Moreover, we remove the 7 predictor variables corresponding to the selected target location in $X$ and eventually we have the data $X \in \mathbb{R}^{814\times 73577}$ and $Y \in \mathbb{R}^{814}$.

This data has also been used in (Ndiaye et al., 2016; Zhang et al., 2020), showing that the sparse group Lasso regularizer is suitable for prediction in climate data. We consider five different target locations for the diversity of experiments. We first set $(w_1, w_2) = (0, 1)$, and select $\lambda$ by CV over (34). In addition, we select $w_1$, $w_2$, and $\lambda$ by CV over (35). Table 4 shows the comparisons of PPDNA, pADMM, dADMM, and S-TISP on the climate data sets with five different target locations. As shown in Table 4, our PPDNA outperforms the other three methods pADMM, dADMM, and S-TISP by a wide margin in term of computational time. In particular, both pADMM and S-TISP fail to solve any instance within 30 minutes, and the accuracy of the solutions returned by S-TISP, approximately $10^{-3}$, are still far from being satisfactory. In addition, we plot the active groups predicting *Air Temperature* in a neighborhood of Dubbo, New South Wales, Australia in Figure 1. As one can expect, the active groups shown in the Figure 1 are close or contiguous to the target location Dubbo.

| target location | $w_1\|\lambda$ | MSE | iter | | | | time | | | | error | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $(N_{\text{train}}, n), g$ | | | PP | pA | dA | ST | PP | pA | dA | ST | PP | pA | dA | ST |
| Dakar | 0.0\|0.501 | 0.294 | 16\|80 | 19469 | 6011 | 35469 | 17 | 30:00 | 05:56 | 30:00 | 1.3e-08 | 1.2e-05 | 9.8e-08 | 5.4e-03 |
| $(15^\circ N, 17^\circ 30' W)$ | 1.0\|1.259 | 0.028 | 19\|109 | 20467 | 17760 | 32782 | 09 | 30:00 | 16:21 | 30:00 | 6.3e-08 | 8.5e-06 | 1.0e-07 | 1.8e-02 |
| $(563, 73577),\ 10511$ | | | | | | | | | | | | | | |
| Dubbo | 0.0\|0.282 | 0.059 | 16\|45 | 19898 | 7245 | 35554 | 47 | 30:00 | 07:14 | 30:00 | 2.3e-08 | 1.5e-05 | 9.9e-08 | 3.8e-03 |
| $(32^\circ 30' S, 147^\circ 30' E)$ | 0.7\|0.562 | 0.021 | 20\|140 | 20235 | 11318 | – | 13 | 30:00 | 11:09 | – | 4.5e-08 | 5.4e-06 | 1.0e-07 | – |
| $(563, 73577),\ 10511$ | | | | | | | | | | | | | | |
| Enshi | 0.0\|0.282 | 0.031 | 1\|186 | 20230 | 7397 | 35177 | 58 | 30:00 | 07:34 | 30:00 | 6.2e-07 | 8.2e-06 | 1.0e-07 | 3.9e-03 |
| $(30^\circ N, 110^\circ E)$ | 0.8\|0.794 | 0.023 | 21\|123 | 20436 | 19163 | – | 08 | 30:00 | 18:39 | – | 5.3e-08 | 9.3e-06 | 1.0e-07 | – |
| $(563, 73577),\ 10511$ | | | | | | | | | | | | | | |
| Weihai | 0.0\|0.282 | 0.045 | 19\|74 | 20220 | 9667 | 31370 | 51 | 30:00 | 09:52 | 30:00 | 1.1e-10 | 3.6e-06 | 1.0e-07 | 4.1e-03 |
| $(37^\circ 30' N, 122^\circ 30' E)$ | 0.9\|0.794 | 0.034 | 20\|123 | 20125 | 21605 | – | 11 | 30:00 | 23:44 | – | 4.7e-08 | 9.0e-06 | 1.0e-07 | – |
| $(563, 73577),\ 10511$ | | | | | | | | | | | | | | |
| Bosilegrad | 0.0\|0.316 | 0.026 | 17\|82 | 20639 | 6783 | 36771 | 37 | 30:00 | 06:32 | 30:00 | 3.6e-08 | 2.9e-06 | 1.0e-07 | 2.7e-03 |
| $(42^\circ 30' N, 22^\circ 30' E)$ | 0.5\|0.447 | 0.024 | 23\|129 | 20267 | 9157 | – | 18 | 30:00 | 09:01 | – | 4.1e-09 | 4.2e-06 | 1.0e-07 | – |
| $(563, 73577),\ 10511$ | | | | | | | | | | | | | | |

Table 4: Square-root sparse group Lasso model on climate data sets with CV over (34) and (35). "–" denotes that the method is not applicable for the instance.

**Gene data** The first data we use is the *colon cancer* data (Alon et al., 1999)[1]. This data has been used in (Li et al., 2017) for the adaptive sparse group Lasso model. It includes

1. It is available at `http://www.weizmann.ac.il/mcb/UriAlon/download/downloadable-data`.

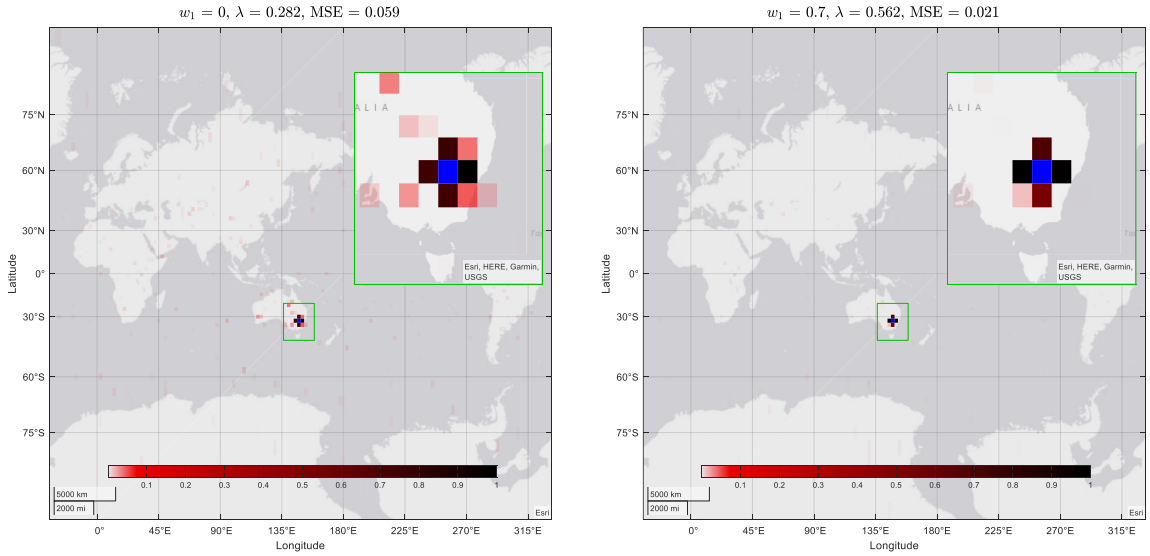


Figure 1: Active groups to predict *Air Temperature* in a neighborhood of Dubbo (in blue). Left: $(w_1, w_2, \lambda) = (0, 1, 0.282)$. Right: $(w_1, w_2, \lambda) = (0.7, 0.3, 0.562)$. At the $l$th location, the value of $\|\beta_{G_l}\|/\max_{1\le l\le g}\|\beta_{G_l}\|$ is displayed.

62 tissues (40 colon tumor tissues and 22 normal tissues), and each tissue $X_i$ includes the expression profiles of 2000 genes. The response variable is assigned according to the label of the tissue: $Y_i = 1$ if $X_i$ is a colon tumor tissue and $Y_i = -1$ otherwise.

In addition, we use the lung cancer data (Monti et al., 2003)[2], which has also been used in (Li et al., 2017). This data includes 197 tissues, and each tissue $X_i$ includes the expression profiles of 1000 genes. Moreover, 197 tissues are divided into four classes: 17 normal, 139 lung adenocarcinoma, 21 squamous cell carcinomas, and 20 carcinoids. Corresponding to the latter three classes, we construct three data sets by letting the binary response variables be labels. Specifically, *lung adenocarcinoma* data set has response variable $Y_i = 1$ if $X_i$ is a lung adenocarcinoma tissue and $Y_i = -1$ otherwise. Similarly, we construct the *squamous cell carcinomas* and *carcinoids* data sets.

Another data tested is the acute leukemia data (Golub et al., 1999)[3]. This data includes 72 samples, and each sample $X_i$ includes the expression profiles of 10713 (repeated) genes. Each sample belongs to one of the three classes: BALL, TALL, or AML. The data sets *BALL*, *TALL*, and *AML* are constructed in the same way as in the lung cancer data above.

In total, we have 7 data sets with binary response variables. For the first 4 data sets, we used the weighted gene co-expression networks (Langfelder and Horvath, 2008) to cluster gene expressions in different groups (modules), and an R package is available for this clustering. The group structures of the last 3 data sets, generated by a similar method, are provided in (Li et al., 2018a). More details about the group structures can be found in Appendix C.

2. It is available at `http://portals.broadinstitute.org/cgi-bin/cancer/publications/view/87`.
3. It is available at `https://github.com/wangyanyanwangyanyan/wangyanyan`.

Table 5 presents the results on real gene data sets, where the parameters $w_1$, $w_2$, and $\lambda$ are selected by CV over (35). For an approximate solution $\beta$, the classification accuracy is computed by $(1 - \text{nnz}\,(\text{sign}\,(X_{\text{test}}\beta) - Y_{\text{test}})/N_{\text{test}}) \cdot 100\%$. For problems with $w_1 = 0$ where S-TISP is applicable, we also include S-TISP in the table. The low MSE and high classification accuracy in Table 5 suggest that the square-root sparse group Lasso model is effective in selecting (groups of) genes related to certain types of tissues and reliable in predicting the class of a tissue; see Appendix C for more details. Moreover, the selected $w_1 \notin \{0, 1\}$ by CV (35) in 4 out of 7 data sets indicates that the sparse group Lasso regularizer can improve the performances of classification and gene selection, compared with the Lasso or group Lasso regularizer. For the *BALL* and *TALL* data sets, pADMM and dADMM return overfitting solutions. Besides, one can observe that the running time for different methods is quite similar, since the gene data sets are of small to medium size and they are pre-processed such that highly irrelevant genes were screened out.

| problem | $w_1\|\lambda$ | MSE | accur | iter | | | | time | | | | error | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $(N_{\text{train}}, n), g$ | | | | PP | pA | dA | ST | PP | pA | dA | ST | PP | pA | dA | ST |
| *colon cancer* $(42, 2000), 11$ | 0.8\|0.631 | 0.448 | 90% | 9\|13 | 2676 | 665 | – | 00 | 00 | 00 | – | 4.9e-08 | 1.7e-08 | 9.5e-08 | – |
| *lung adenocarcinoma* $(132, 1000), 8$ | 0.0\|0.562 | 0.156 | 97% | 9\|13 | 717 | 613 | 5347 | 00 | 00 | 00 | 00 | 5.7e-09 | 1.4e-08 | 9.9e-08 | 1.0e-07 |
| *squamous cell carcinomas* $(131, 1000), 8$ | 0.0\|1.259 | 0.131 | 97% | 10\|15 | 1861 | 690 | 2783 | 00 | 00 | 00 | 00 | 5.4e-09 | 3.7e-08 | 9.6e-08 | 1.0e-07 |
| *carcinoids* $(131, 1000), 8$ | 0.9\|1.000 | 0.008 | 100% | 9\|12 | 1076 | 509 | – | 00 | 00 | 00 | – | 4.9e-09 | 2.6e-08 | 9.6e-08 | – |
| *BALL* $(48, 10713), 42$ | 0.0\|0.200 | 0.141 | 100% | 10\|19 | 453 | 404 | 1596 | 00 | 00 | 00 | 02 | 1.6e-10 | ∗9.9e-08 | ∗4.9e-08 | #2.1e-13 |
| *TALL* $(48, 10713), 42$ | 0.9\|0.891 | 0.126 | 96% | 11\|25 | 1400 | 2342 | – | 01 | 02 | 03 | – | 3.9e-11 | ∗9.1e-08 | ∗1.0e-07 | – |
| *AML* $(48, 10713), 42$ | 0.9\|1.585 | 0.193 | 100% | 12\|37 | 18393 | 1755 | – | 00 | 29 | 02 | – | 2.4e-08 | 2.1e-08 | 5.7e-08 | – |

Table 5: Square-root sparse group Lasso model on gene data sets with CV over (35). "–" denotes that the method is not applicable for the instance. The errors reported are $\Delta_{\text{kkt}}$, $*\Delta_{\text{pd.gap}}$, and $\#\Delta_{\text{var.gap}}$, given by (26), (27), and (28), respectively.

### 4.4 Comparison of Efficiency for Solving the Square-root Fused Lasso Problem

In this section, we compare PPDNA, pADMM, and dADMM for solving the square-root fused Lasso problem when the regularizer $p$ is the fused Lasso regularizer (7). We found that a framework of ADMM was applied in (Jiang et al., 2021) for solving the square-root fused Lasso problem. However, it seems that they applied ADMM in a non-rigorous way as their formulation of the augmented Lagrangian function might not be correct. Except (Jiang et al., 2021), there is currently no solver for solving the square-root fused Lasso problem, to the best of our knowledge. For all tables in this section, we denote PPDNA, pADMM, and dADMM by "PP", "pA", and "dA", respectively.

#### 4.4.1 UCI Data

Again, we use UCI data sets *housing*, *bodyfat*, *pyrim*, and *triazines* described in Section 4.3.1. We choose $w_1 = w_2 = 0.5$ and $\lambda \in \{1, 5, \lambda_{\text{Jia}}\}$. See Appendix B for the details of $\lambda_{\text{Jia}}$ given in (Jiang et al., 2021). The numerical results of PPDNA, pADMM, and dADMM are reported in Table 6. Table 6 shows that our PPDNA substantially outperforms both pADMM and dADMM for solving the square-root fused Lasso problem on the UCI data sets. In particular, our PPDNA takes less than 1 minute for all instances; while pADMM fails to return accurate solutions within 30 minutes for more than half of the instances. One can conclude that our algorithm is efficient for solving the square-root fused Lasso problem on the UCI data sets.

#### 4.4.2 Real Data

In this section, we test the square-root fused Lasso model on four real data sets used in (Jiang et al., 2021). The *inbred mouse*[4] data includes 60 samples, where each sample includes 22689 genes, and the response variable is the feature measured by stearoyl-coenzyme desaturase 1 with probe set ID given as 1415965_at. The *rat eye*[5] data includes 120 rats samples, where each sample includes 31098 gene probes, and the response variable is selected with respect to 1389163_at. The *credit card*[6] data includes 284807 transactions samples in which 492 samples are labeled as frauds, and each sample includes 29 features. The *safe driver*[7] data includes 595212 car insurance observations, where each observation includes 57 features. We refer the readers to (Jiang et al., 2021, Section 4.3) for detailed descriptions of the data.

Again, we choose $w_1 = w_2 = 0.5$ and $\lambda \in \{1, 5, \lambda_{\text{Jia}}\}$ with $\lambda_{\text{Jia}}$ given in (Jiang et al., 2021). The comparisons of PPDNA, pADMM, and dADMM for solving the square-root fused Lasso model on the four real data sets are reported in Table 7. On can observe that our PPDNA significantly outperforms both pADMM and dADMM for all instances. In particular, both pADMM and dADMM fail to solve the problem *safe driver* within 30 minutes.

### 4.5 Simulations with Varying Noise Levels

Here we are interested in how the noise level $\sigma$ in (1) is associated with the tuning parameter $\lambda$ in the least-square model (2) and the square-root model (3). In particular, we let $p$ be the sparse group Lasso regularizer (6). In this experiment, the noise level $\sigma$ is chosen from the set $\{10^{-1}, 10^{-0.8}, \ldots, 10^{1}\}$, the data is simulated from **Example 3** in Section 4.3.1 with $N = 100, g = 50, n = 150$, and the parameters $w_1$ and $w_2$ are set to be 0.5. We regard the 8-fold CV selected tuning parameters $\lambda_{(2)}$ and $\lambda_{(3)}$ as the optimal parameters for model (2) and model (3), respectively. We repeat the experiments 100 times, and we plot $\lambda$ and MSE against $\sigma$ in Figure 2.

We can observe from the left panel of Figure 2 that as $\sigma$ varies from $10^{-1}$ to $10^{1}$, the curve of $\lambda_{(3)}$ roughly remains flat with the values of $\lambda_{(3)}$ staying in the small range $[10^{-0.5}, 10^{0.5}]$. In contrast, $\lambda_{(2)}$ shown in the left panel of Figure 2 varies wildly from $10^{-1}$

4. It is available at `https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE3330`.
5. It is available at `https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE5680`.
6. It is available at `https://www.kaggle.com/mlg-ulb/creditcardfraud`.
7. It is available at `https://www.kaggle.com/c/porto-seguro-safe-driver-prediction/data`.

| problem $(N, n)$ | $\lambda$ | nnz\|nnzB | iter PP | iter pA | iter dA | time PP | time pA | time dA | error PP | error pA | error dA |
|---|---|---|---|---|---|---|---|---|---|---|---|
| *housing* (253, 77520) | 1.000 | 336\|129 | 19\|119 | 35001 | 13873 | 05 | 25:01 | 05:48 | 5.2e-08 | 8.4e-08 | 1.0e-07 |
| | 5.000 | 168\|28 | 16\|81 | 42814 | 30001 | 03 | 30:00 | 11:40 | 2.6e-08 | 2.1e-06 | 5.1e-08 |
| | 9.282 | 141\|18 | 14\|53 | 42015 | 32501 | 02 | 30:00 | 12:22 | 4.0e-08 | 4.7e-07 | 7.5e-08 |
| *bodyfat* (126, 116280) | 1.000 | 295\|63 | 24\|121 | 27501 | 6125 | 05 | 15:25 | 02:05 | 2.2e-08 | 9.5e-08 | 1.0e-07 |
| | 5.000 | 132\|12 | 14\|56 | 45001 | 27501 | 02 | 25:36 | 08:34 | 4.3e-09 | 9.1e-08 | 6.8e-08 |
| | 8.969 | 79\|6 | 13\|41 | 54052 | 35001 | 01 | 30:00 | 10:29 | 2.3e-09 | 3.8e-07 | 8.1e-08 |
| *pyrim* (37, 169911) | 1.000 | 535\|69 | 40\|169 | 67611 | 116842 | 08 | 30:00 | 30:00 | 4.3e-07 | ∗8.5e-05 | 7.8e-01 |
| | 5.000 | 386\|14 | 14\|52 | 67844 | 20001 | 02 | 30:00 | 05:10 | 4.5e-08 | 8.8e-02 | 3.5e-08 |
| | 7.848 | 669\|10 | 15\|69 | 60001 | 24828 | 03 | 26:31 | 06:17 | 6.1e-10 | 2.6e-08 | 2.5e-08 |
| *triazines* (93, 557845) | 1.000 | 2771\|165 | 34\|247 | 8176 | 15422 | 43 | 30:00 | 30:00 | 5.8e-08 | 1.3e-05 | 3.1e-03 |
| | 5.000 | 1310\|57 | 22\|117 | 8122 | 15399 | 22 | 30:00 | 30:00 | 8.9e-08 | 8.6e-05 | 1.2e-04 |
| | 9.278 | 196\|12 | 13\|37 | 8053 | 15716 | 10 | 30:00 | 30:00 | 1.9e-08 | 5.7e-04 | 8.2e-04 |

Table 6: Square-root fused Lasso model on UCI data sets with $w_1 = w_2 = 0.5$. The errors reported are $\Delta_{\text{kkt}}$ and $*\Delta_{\text{pd.gap}}$, given by (26) and (27), respectively.

| problem $(N, n)$ | $\lambda$ | nnz\|nnzB | iter PP | iter pA | iter dA | time PP | time pA | time dA | error PP | error pA | error dA |
|---|---|---|---|---|---|---|---|---|---|---|---|
| *inbred mouse* (30, 22689) | 1.000 | 85\|4 | 11\|21 | 2231 | 6626 | 02 | 05 | 12 | 2.9e-08 | 1.1e-08 | 8.8e-08 |
| | 5.000 | 1347\|1 | 13\|42 | 2043 | 779 | 00 | 04 | 01 | 1.2e-08 | 5.8e-10 | 8.6e-08 |
| | 6.913 | 3092\|1 | 13\|42 | 2009 | 601 | 00 | 05 | 01 | 2.4e-09 | 1.1e-09 | 7.4e-08 |
| *rat eye* (60, 31098) | 1.000 | 433\|10 | 12\|30 | 1861 | 2200 | 01 | 09 | 09 | 1.3e-08 | 2.2e-08 | 1.0e-07 |
| | 5.000 | 3303\|4 | 12\|40 | 1549 | 1305 | 00 | 09 | 05 | 4.4e-08 | 3.4e-08 | 1.0e-07 |
| | 7.798 | 7297\|4 | 12\|35 | 1549 | 666 | 00 | 08 | 02 | 2.6e-08 | 7.7e-08 | 1.0e-07 |
| *credit card* (142404, 29) | 1.000 | 27\|26 | 11\|24 | 5485 | 1343 | 02 | 42 | 21 | 3.5e-09 | 7.3e-08 | 8.1e-08 |
| | 5.000 | 25\|24 | 16\|84 | 15001 | 5210 | 05 | 01:49 | 01:19 | 6.1e-08 | 3.2e-08 | 6.9e-08 |
| | 5.683 | 25\|24 | 18\|96 | 15001 | 9953 | 06 | 01:52 | 02:28 | 3.2e-08 | 2.7e-08 | 7.6e-08 |
| *safe driver* (595212, 57) | 1.000 | 48\|35 | 31\|223 | 35850 | 14429 | 01:29 | 30:00 | 30:00 | 2.7e-08 | 1.9e-05 | 1.3e-04 |
| | 5.000 | 48\|30 | 23\|134 | 36352 | 14333 | 39 | 30:00 | 30:00 | 2.8e-08 | 5.3e-07 | 5.4e-05 |
| | 6.242 | 47\|28 | 24\|133 | 35173 | 14308 | 37 | 30:00 | 30:00 | 1.9e-08 | 4.6e-07 | 1.0e-05 |

Table 7: Square-root fused Lasso model on real data sets with $w_1 = w_2 = 0.5$.

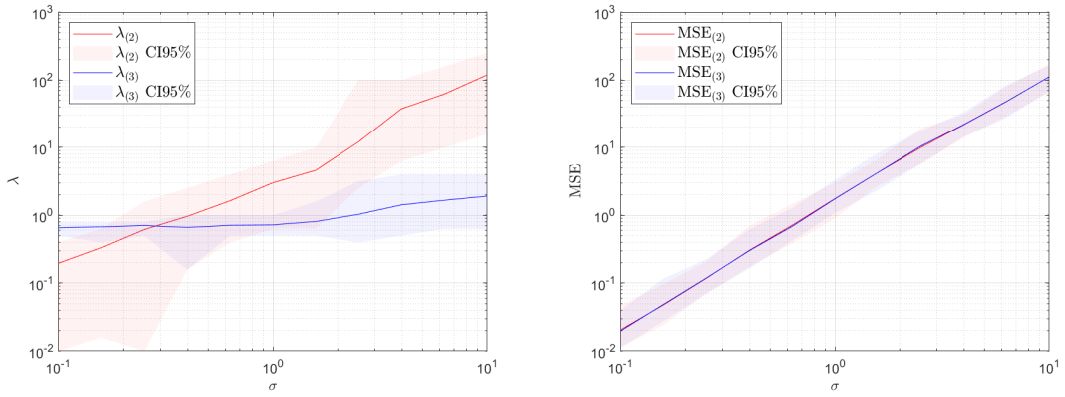


Figure 2: The mean of (left) $\lambda$ and (right) MSE, and the 95% confidence interval (CI).

to $10^2$. This phenomenon verifies numerically the advantage of the square-root model (3) in terms of choosing the tuning parameters compared with the least-square model (2), namely, the parameter $\lambda$ in the square-root model (3) can be tuned independent of the noise level $\sigma$. In addition, it can be observed from the slope of the MSE curves in the right panel of Figure 2 that both the MSE achieved by (2) and that by (3) are approximately quadratic in the noise level $\sigma$.

## 5. Conclusion

In this paper, we have given a unified proof to show that any square-root regularized model whose penalty function being the sum of a simple norm and a seminorm can be interpreted as the distributionally robust optimization formulation of the corresponding least-squares problem. For solving a generic square-root regularized model, we have developed a proximal point dual semismooth Newton algorithmic framework to efficiently solve the resulting convex minimization problem whose objective is the sum of two nonsmooth terms corresponding to the square-root loss and the regularizer respectively. We have illustrated that the general framework can be adopted to solve the square-root sparse group Lasso and the square-root fused Lasso models. Our extensive numerical experiments have shown that the proposed algorithm is indeed highly efficient for solving the square-root sparse group Lasso and the square-root fused Lasso models, as compared to popular first order methods based on the ADMM framework.

## Acknowledgments

The research of Kim-Chuan Toh is supported by the Ministry of Education, Singapore, under its Academic Research Fund Tier 3 grant call (MOE-2019-T3-1-010). The research of Yangjing Zhang is supported by the National Natural Science Foundation of China under grant number 12201617.

## Appendix A. Remark on Proposition 2

Proposition 2 includes a similar result in (Maurer and Pontil, 2012). For simplicity, we show that a simplified form of their results can be derived from Proposition 2. In (Maurer and Pontil, 2012), $\mathcal{M} := \{M_{(l)}\}_{1\le l\le g}$ denotes a set of symmetric matrices $M_{(l)} \in \mathbb{S}^n$, and the operator $\|\cdot\|_{\mathcal{M}} : \mathbb{R}^n \to [0,+\infty]$ is defined by

$$\|\alpha\|_{\mathcal{M}} := \inf_{\alpha_{(1)},\ldots,\alpha_{(g)}} \left\{ \sum_{l=1}^{g} \|\alpha_{(l)}\| \,\middle|\, \alpha_{(l)} \in \mathbb{R}^n,\ \sum_{l=1}^{g} M_{(l)}\alpha_{(l)} = \alpha \right\} \ \forall\, \alpha \in \mathbb{R}^n. \tag{36}$$

It was shown in (Maurer and Pontil, 2012) that $\|\cdot\|_{\mathcal{M}}$ is indeed a norm on the subspace of $\mathbb{R}^n$ where it is finite, and the dual norm is given by

$$\|\beta\|_{\mathcal{M}^*} := \sup_{1\le l\le g} \left\{ \|M_{(l)}\beta\| \right\} \ \forall\, \beta \in \mathbb{R}^n. \tag{37}$$

We define $B := \big(M_{(1)},\ldots,M_{(g)}\big)^T \in \mathbb{R}^{ng\times n}$ and $Q(\bar{\alpha}) := \sup_{1\le l\le g}\left\{\|\alpha_{(l)}\|\right\}$ for any $\bar{\alpha} = (\alpha_{(1)},\ldots,\alpha_{(g)}) \in \mathbb{R}^n \times \cdots \times \mathbb{R}^n$. $Q$ is a norm on $\mathbb{R}^{ng}$, and its dual norm is given by

$Q_*(\bar{\alpha}) := \sum_{l=1}^g \|\alpha_{(l)}\| \ \forall\, \bar{\alpha} = (\alpha_{(1)}, \ldots, \alpha_{(g)}) \in \mathbb{R}^n \times \cdots \times \mathbb{R}^n$. We can see from (36) that $\|\cdot\|_{\mathcal{M}}$ coincides with $p_*$ in (15), i.e., $\|\alpha\|_{\mathcal{M}} = \inf_{\bar{\alpha}}\{Q_*(\bar{\alpha}) \,|\, B^T\bar{\alpha} = \alpha\} \ \forall\, \alpha \in \mathbb{R}^n$. It is indeed a norm on $\texttt{Range}(B^T)$ from Proposition 2(a). We can also see from (37) that $\|\cdot\|_{\mathcal{M}^*}$ coincides with $p$ in (14), i.e., $\|\beta\|_{\mathcal{M}^*} = Q(B\beta) \ \forall\, \beta \in \mathbb{R}^n$. Therefore, we can derive the result that $\|\cdot\|_{\mathcal{M}}$ and $\|\cdot\|_{\mathcal{M}^*}$ are dual to each other directly from Proposition 2.

## Appendix B. Theoretical Parameter $\lambda$

The independence of the tuning parameter $\lambda$ on the unknown noise level $\sigma$ is one of the nice statistical properties of the square-root regularized model. In this section, we present various selections of $\lambda$ which has been studied in literature for the problem (3). The cumulative distribution function of the standard normal distribution $\mathcal{N}(0,1)$ is denoted by $\Phi(x) := \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-t^2/2} dt$. The cumulative distribution function of the $F$-distribution with the degrees of freedom $a$ and $b$ is denoted by $\mathcal{F}_{a,b}$. The quantile function with respect to a cumulative distribution function $F$ is denoted by $\mathcal{Q}_F(a) := \inf\{x \in \mathbb{R} \mid a \le F(x)\}$.

**Sparse group Lasso regularizer** (6) $p(\beta) = w\|\beta\|_1 + (1-w)\sum_{l=1}^g \sqrt{|G_l|}\|\beta_{G_l}\| \ \forall\, \beta \in \mathbb{R}^n$, where $w \in [0,1]$. We summarize the selections of $\lambda$ which are independent on $\sigma$ in (Belloni et al., 2011; Bunea et al., 2013; Stucky and van de Geer, 2017; Blanchet et al., 2019a; Blanchet and Kang, 2017). In the following formulations, $a$ is chosen to be 0.05. When $N < 10^4$, we calculate $\mathrm{E}[X^TX] \approx \frac{1}{N}X^TX$; otherwise, we randomly sample $10^4$ predictor vectors from $X$ to form $\tilde{X}$, and estimate $\mathrm{E}[X^TX] \approx \frac{1}{10^4}\tilde{X}^T\tilde{X}$.

1. When $w = 1$, $\lambda$ can be selected from $\Lambda_S := \{\lambda_{\mathrm{Bel}}, \lambda_{\mathrm{StS}}, \lambda_{\mathrm{BlS}}\}$.
   - (Belloni et al., 2011) $\lambda_{\mathrm{Bel}} := 1.1\Phi^{-1}\left(1 - \frac{a}{2n}\right)$.
   - (Stucky and van de Geer, 2017) Denote $t := \sqrt{\log\left(\frac{4}{a}\right)}, \Delta := \sqrt{1 - t\sqrt{\frac{4}{N}}}$, then $\lambda_{\mathrm{StS}} := \sqrt{2}\frac{t}{\Delta} + \sqrt{2}\left(2 + \sqrt{\log(n)}\right)$.
   - (Blanchet et al., 2019a) Estimate $Z \sim \mathcal{N}(0, \mathrm{E}[X^TX])$ and $F$ as the cumulative distribution of $\frac{\pi}{\pi-2}\|Z\|_\infty^2$. Let $\hat{\eta}_{1-a} := \mathcal{Q}_F(1-a)$, and $\lambda_{\mathrm{BlS}} := \sqrt{\hat{\eta}_{1-a}}$.
2. When $w = 0$, $\lambda$ can be selected from $\Lambda_G := \{\lambda_{\mathrm{Bun}}, \lambda_{\mathrm{StG}}, \lambda_{\mathrm{BlG}}\}$.
   - (Bunea et al., 2013) Denote $T_{\max} := \max_{1\le l\le g}\{|G_l|\}, T_{\min} := \min_{1\le l\le g}\{|G_l|\}$, $\tau_0 := \mathcal{F}^{-1}_{T_{\min}, N-T_{\min}}\left(1 - \frac{a}{g}\right)$ and $\zeta_{\max} := \max\left\{\frac{1}{N}\|X_{G_l}\|^2 \mid 1 \le l \le g\right\}$. Suppose that $T_{\min}\tau_0 + N - T_{\max} > 0$, then $\lambda_{\mathrm{Bun}} := \sqrt{\frac{\zeta_{\max}\tau_0}{T_{\min}\tau_0 + N - T_{\max}}}\sqrt{N}$.
   - (Stucky and van de Geer, 2017) Denote $t := \sqrt{\log\left(\frac{4}{a}\right)}, \Delta := \sqrt{1 - t\sqrt{\frac{4}{N}}}$, then $\lambda_{\mathrm{StG}} := \sqrt{2}\frac{t}{\Delta} + \sqrt{2}\left(2 + \sqrt{\log(g)}\right)$.
   - (Blanchet and Kang, 2017) Estimate $Z \sim \mathcal{N}(0, \mathrm{E}[X^TX])$ and $F$ as the cumulative distribution of $\frac{\pi}{\pi-2}(p_*(Z))^2$. Let $\hat{\eta}_{1-a} := \mathcal{Q}_F(1-a)$, and $\lambda_{\mathrm{BlG}} := \sqrt{\hat{\eta}_{1-a}}$.

**Fused Lasso regularizer** (7) Following (Jiang et al., 2021), we let $a = 0.05$, $t = \sqrt{4\log(\frac{1}{a})/N + 4\log(\frac{1}{a})/N}$ and calculate $\lambda_{\mathrm{Jia}} = 2.2\sqrt{2\log(n)/(1+t)}$.

## Appendix C. Gene Data Sets

In this section, we provide more details about the gene data sets used in Section 4.3.3 and their numerical results.

| black | blue | brown | green | grey | magenta | pink | purple | red | turquoise | yellow |
|---|---|---|---|---|---|---|---|---|---|---|
| 71 | 420 | 367 | 98 | 18 | 63 | 64 | 43 | 87 | 507 | 262 |

Table 8: The number of genes in each module in the colon cancer data.

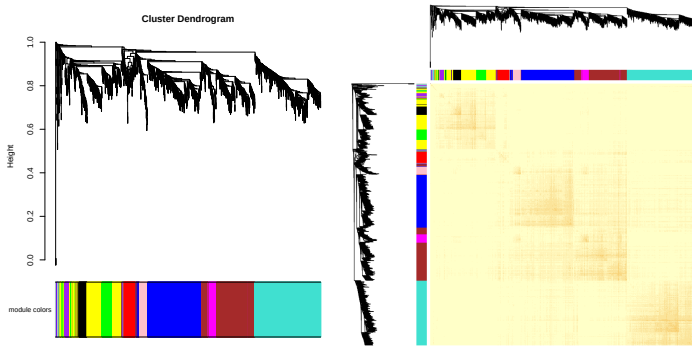


Figure 3: Left: clustering dendrogram of 2000 gene profiles from the colon cancer patients. Right: heatmap of gene-gene connectivity.

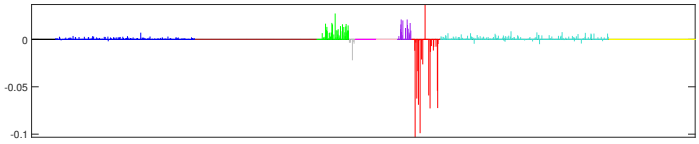


Figure 4: The value of $\beta$ (coordinates reordered by its groups) in colon cancer data.

***Colon cancer data*** This data includes 62 tissues with binary labels, and each tissue includes the expression profiles of 2000 genes. We use the R package WGCNA (Langfelder and Horvath, 2008) with the power parameter 6 to construct the weighted gene co-expression networks, and divide 2000 genes into 11 modules which are marked by 11 colors in Figure 3. The number of genes in each module is given in Table 8. The value of the approximate solution $\beta$ (coordinates reordered by its groups) with CV (35) selected parameters is presented in Figure 4. One can observe that there are 6 active groups out of total 11 groups, which may imply that the red, green, and purple groups are key groups to classify the colon cancer patients. Moreover, we can see that $\beta$ is also sparse within each group, which indicates the effectiveness of the square-root sparse group Lasso regularizer.

***Lung cancer data*** This data includes 197 tissues, and each tissue includes the expression profiles of 1000 genes. We use the R package WGCNA (Langfelder and Horvath, 2008) with the power parameter 6 to construct the weighted gene co-expression networks,

| black | blue | brown | green | grey | red | turquoise | yellow |
|---|---|---|---|---|---|---|---|
| 28 | 186 | 112 | 58 | 172 | 38 | 332 | 74 |

Table 9: The number of genes in each module in the lung cancer data.

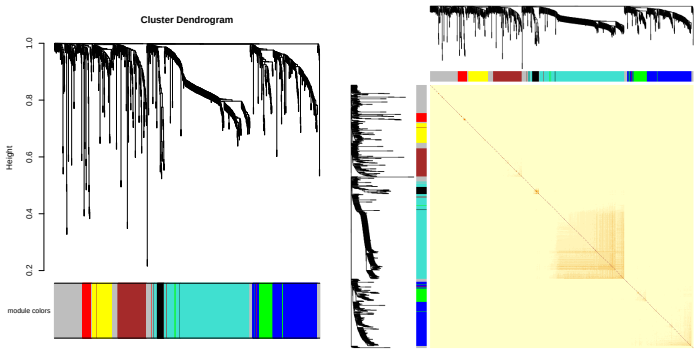


Figure 5: Left: clustering dendrogram of 1000 gene profiles from the lung cancer patients. Right: heatmap of gene-gene connectivity.

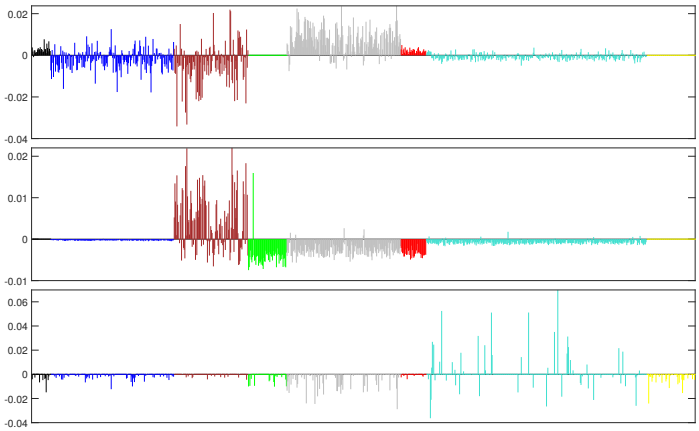


Figure 6: The value of $\beta$ (coordinates reordered by its groups) in lung cancer data. Top: *lung adenocarcinoma*, middle: *squamous cell carcinomas*, bottom: *carcinoids*.

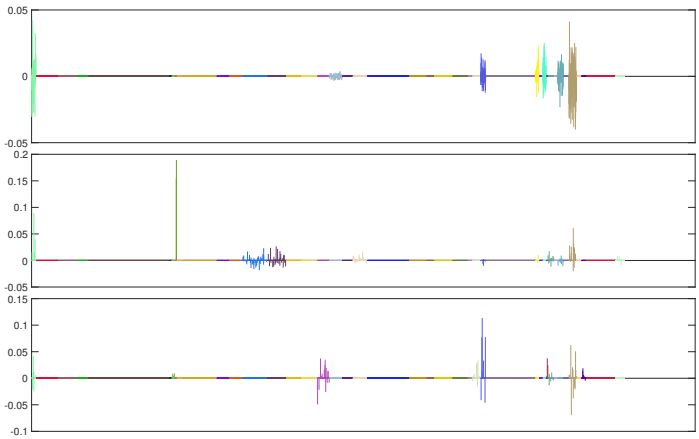


Figure 7: The value of $\beta$ (coordinates reordered by its groups) for acute leukemia data. Top: *BALL*, middle: *TALL*, bottom: *AML*.

and divide 1000 genes into 8 modules which are marked by 8 colors in Figure 5. The number of genes in each module is given in Table 9. The value of the approximate solution $\beta$ (coordinates reordered by its groups) with CV (35) selected parameters in *lung adenocarcinoma*, *squamous cell carcinomas*, and *carcinoids* data sets is presented in Figure 6. One can observe that the set of selected groups varies for different data sets. For example, the brown group is one of the key modules to classify lung adenocarcinoma and squamous cell carcinomas tissues, while the turquoise module is preferred in the carcinoids data set. Note that (35) returns $w_1 = 0$ in the first two data sets and $w_1 = 0.9$ in the last data set, which results in the sparsity of the last regression vector.

***Acute leukemia data*** The raw acute leukemia data set includes 72 samples of 3571 gene expressions. Following (Li et al., 2018a), the grouping strategy with repeated genes was applied to get the data with 10713 gene expressions. The value of the approximate solution $\beta$ (coordinates reordered by its groups) with CV (35) selected parameters in *BALL*, *TALL*, and *AML* data sets is presented in Figure 7. Similar conclusions could be made as in the lung cancer data above.

## References

U. Alon, N. Barkai, D. A. Notterman, K. Gish, S. Ybarra, D. Mack, and A. J. Levine. 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.

D. Angelosante and G. B. Giannakis. RLS-weighted Lasso for adaptive estimation of sparse signals. In *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 3245–3248, 2009.

A. Asuncion and D. Newman. UCI machine learning repository, 2007.

J. A. Bazerque, G. Mateos, and G. B. Giannakis. Group-Lasso on splines for spectrum cartography. *IEEE Transactions on Signal Processing*, 59(10):4648–4663, 2011.

S. R. Becker, E. J. Candès, and M. C. Grant. Templates for convex cone problems with applications to sparse signal recovery. *Mathematical Programming Computation*, 3(3): 165–218, 2011.

A. Belloni, V. Chernozhukov, and L. Wang. Square-root Lasso: pivotal recovery of sparse signals via conic programming. *Biometrika*, 98(4):791–806, 2011.

D. Bertsimas and M. S. Copenhaver. Characterization of the equivalence of robustification and regularization in linear and matrix regression. *European Journal of Operational Research*, 270(3):931–942, 2018.

P. J. Bickel, Y. Ritov, and A. B. Tsybakov. Simultaneous analysis of Lasso and Dantzig selector. *The Annals of Statistics*, 37(4):1705–1732, 2009.

J. Bigot and T. Klein. Characterization of barycenters in the Wasserstein space by averaging optimal transport maps. *ESAIM: Probability and Statistics*, 22:35–57, 2018.

J. Blanchet and Y. Kang. Distributionally robust groupwise regularization estimator. In *Proceedings of the Ninth Asian Conference on Machine Learning*, volume 77, pages 97–112, 2017.

J. Blanchet and K. Murthy. Quantifying distributional model risk via optimal transport. *Mathematics of Operations Research*, 44(2):565–600, 2019.

J. Blanchet, Y. Kang, and K. Murthy. Robust Wasserstein profile inference and applications to machine learning. *Journal of Applied Probability*, 56(3):830–857, 2019a.

J. Blanchet, Y. Kang, K. Murthy, and F. Zhang. Data-driven optimal transport cost selection for distributionally robust optimization. In *Proceedings of the Winter Simulation Conference*, pages 3740–3751, 2019b.

M. Bogdan, E. Van Den Berg, C. Sabatti, W. Su, and E. J. Candès. SLOPE-adaptive variable selection via convex optimization. *The Annals of Applied Statistics*, 9(3):1103–1140, 2015.

F. Bunea, J. Lederer, and Y. She. The group square-root Lasso: Theoretical properties and fast algorithms. *IEEE Transactions on Information Theory*, 60(2):1313–1325, 2013.

C.-C. Chang and C.-J. Lin. Libsvm: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3):1–27, 2011.

L. Chen, D. F. Sun, and K.-C. Toh. An efficient inexact symmetric Gauss–Seidel based majorized ADMM for high-dimensional convex composite conic programming. *Mathematical Programming*, 161(1-2):237–270, 2017.

M. Cuturi and A. Doucet. Fast computation of Wasserstein barycenters. In *Proceedings of the 31st International Conference on Machine Learning*, volume 32, pages 685–693, 2014.

L. El Ghaoui and H. Lebret. Robust solutions to least-squares problems with uncertain data. *SIAM Journal on Matrix Analysis and Applications*, 18(4):1035–1064, 1997.

M. Fazel, T. K. Pong, D. F. Sun, and P. Tseng. Hankel matrix rank minimization with applications to system identification and realization. *SIAM Journal on Matrix Analysis and Applications*, 34(3):946–977, 2013.

J. Friedman, T. Hastie, and R. Tibshirani. A note on the group Lasso and a sparse group Lasso. *arXiv preprint arXiv:1001.0736*, 2010.

D. Gabay and B. Mercier. A dual algorithm for the solution of nonlinear variational problems via finite element approximation. *Computers and Mathematics with Applications*, 2(1): 17–40, 1976.

R. Glowinski and A. Marroco. Sur l'approximation, par éléments finis d'ordre un, et la résolution, par pénalisation-dualité d'une classe de problèmes de Dirichlet non linéaires. *Revue française d'automatique, informatique, recherche opérationnelle. Analyse numérique*, 9(R2):41–76, 1975.

G. H. Golub and C. F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, Baltimore, MD, fourth edition, 2013.

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

L. Huang, J. Jia, B. Yu, B.-G. Chun, P. Maniatis, and M. Naik. Predicting execution time of computer programs using sparse polynomial regression. *Advances in Neural Information Processing Systems*, 23:883–891, 2010.

L. Jacob, G. Obozinski, and J.-P. Vert. Group Lasso with overlap and graph Lasso. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 433–440, 2009.

H. Jiang, S. Luo, and Y. Dong. Simultaneous feature selection and clustering based on square root optimization. *European Journal of Operational Research*, 289(1):214–231, 2021. ISSN 0377-2217.

E. Kalnay, M. Kanamitsu, R. Kistler, W. Collins, D. Deaven, L. Gandin, M. Iredell, S. Saha, G. White, J. Woollen, Y. Zhu, M. Chelliah, W. Ebisuzaki, W. Higgins, J. Janowiak, K. C. Mo, C. Ropelewski, J. Wang, A. Leetmaa, R. Reynolds, R. Jenne, and D. Joseph. The NCEP/NCAR 40-year reanalysis project. *Bulletin of the American Meteorological Society*, 77(3):437–472, 1996.

P. Langfelder and S. Horvath. WGCNA: an R package for weighted correlation network analysis. *BMC Bioinformatics*, 9(1):1–13, 2008.

J. Li and J. Z. Wang. Real-time computerized annotation of pictures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(6):985–1002, 2008.

J. Li, W. Dong, and D. Meng. Grouped gene selection of cancer via adaptive sparse group lasso based on conditional mutual information. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 15(6):2028–2038, 2017.

J. Li, Y. Wang, T. Jiang, H. Xiao, and X. Song. Grouped gene selection and multi-classification of acute leukemia via new regularized multinomial regression. *Gene*, 667: 18–24, 2018a.

X. Li, T. Zhao, X. Yuan, and H. Liu. The flare package for high dimensional linear regression and precision matrix estimation in R. *Journal of Machine Learning Research*, 16(18):553–557, 2015.

X. Li, D. F. Sun, and K.-C. Toh. On efficiently solving the subproblems of a level-set method for fused Lasso problems. *SIAM Journal on Optimization*, 28(2):1842–1866, 2018b.

X. Li, D. F. Sun, and K.-C. Toh. A highly efficient semismooth Newton augmented Lagrangian method for solving Lasso problems. *SIAM Journal on Optimization*, 28(1): 433–458, 2018c.

X. Li, H. Jiang, J. Haupt, R. Arora, H. Liu, M. Hong, and T. Zhao. On fast convergence of proximal algorithms for SQRT-lasso optimization: Don't worry about its nonsmooth loss function. In *Uncertainty in Artificial Intelligence*, pages 49–59. PMLR, 2020.

M. Lin, Y.-J. Liu, D. F. Sun, and K.-C. Toh. Efficient sparse semismooth Newton methods for the clustered Lasso problem. *SIAM Journal on Optimization*, 29(3):2026–2052, 2019.

K. Lounici, M. Pontil, S. van de Geer, and A. B. Tsybakov. Oracle inequalities and optimal inference under group sparsity. *The Annals of Statistics*, 39(4):2164–2204, 2011.

Z. Luo, D. F. Sun, K.-C. Toh, and N. Xiu. Solving the OSCAR and SLOPE models using a semismooth Newton-based augmented Lagrangian method. *Journal of Machine Learning Research*, 20(106):1–25, 2019.

A. Maurer and M. Pontil. Structured sparsity and generalization. *Journal of Machine Learning Research*, 13(23):671–690, 2012.

N. Meinshausen and B. Yu. Lasso-type recovery of sparse representations for high-dimensional data. *The Annals of Statistics*, 37(1):246–270, 2009.

S. Monti, P. Tamayo, J. Mesirov, and T. Golub. Consensus clustering: a resampling-based method for class discovery and visualization of gene expression microarray data. *Machine Learning*, 52(1):91–118, 2003.

J.-J. Moreau. Proximité et dualité dans un espace hilbertien. *Bulletin de la Société Mathématique de France*, 93(2):273–299, 1965.

R. Muthukrishnan and R. Rohini. Lasso: A feature selection technique in predictive modeling for machine learning. In *2016 IEEE International Conference on Advances in Computer Applications (ICACA)*, pages 18–20, 2016.

E. Ndiaye, O. Fercoq, A. Gramfort, and J. Salmon. Gap safe screening rules for sparse-group lasso. In *Advances in Neural Information Processing Systems*, pages 388–396, 2016.

S. Petry, C. Flexeder, and G. Tutz. Pairwise fused Lasso. Technical Report 102, Department of Statistics, University of Munich, Munich, 2011.

J. Rabin, G. Peyré, J. Delon, and M. Bernot. Wasserstein barycenter and its application to texture mixing. In *Scale Space and Variational Methods in Computer Vision*, volume 6667, pages 435–446, 2011.

N. Rao, R. Nowak, C. Cox, and T. Rogers. Classification with the sparse group Lasso. *IEEE Transactions on Signal Processing*, 64(2):448–463, 2015.

R. T. Rockafellar. *Convex Analysis*. Princeton University Press, Princeton, NJ, 1970.

R. T. Rockafellar. Monotone operators and the proximal point algorithm. *SIAM Journal on Control and Optimization*, 14(5):877–898, 1976.

S. Shafieezadeh-Abadeh, P. M. Esfahani, and D. Kuhn. Distributionally robust Logistic regression. In *Proceedings of the 28th International Conference on Neural Information Processing Systems*, volume 1, pages 1576–1584, 2015.

Y. She. Sparse regression with exact clustering. *Electronic Journal of Statistics*, 4:1055–1096, 2010.

M. Sion. On general minimax theorems. *Pacific Journal of Mathematics*, 8:171–176, 1958.

B. Stucky and S. van de Geer. Sharp oracle inequalities for square root regularization. *Journal of Machine Learning Research*, 18(67):1–29, 2017.

P. Tang, C. Wang, D. F. Sun, and K.-C. Toh. A sparse semismooth Newton based proximal majorization-minimization algorithm for nonconvex square-root-loss regression problems. *Journal of Machine Learning Research*, 21(226):1–38, 2020.

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

R. Tibshirani, M. Saunders, S. Rosset, J. Zhu, and K. Knight. Sparsity and smoothness via the fused Lasso. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(1):91–108, 2005.

K.-C. Toh, M. J. Todd, and R. H. Tütüncü. SDPT3 – A MATLAB software package for semidefinite programming, version 1.3. *Optimization Methods and Software*, 11(1-4): 545–581, 1999.

R. H. Tütüncü, K.-C. Toh, and M. J. Todd. Solving semidefinite-quadratic-linear programs using SDPT3. *Mathematical Programming*, 95(2):189–217, 2003.

S. van de Geer. Weakly decomposable regularization penalties and structured sparsity. *Scandinavian Journal of Statistics*, 41(1):72–86, 2014.

C. Villani. *Optimal Transport: Old and New*, volume 338. Springer Berlin Heidelberg, 2008.

W. Xie and H. Yang. The structured smooth adjustment for square-root regularization: Theory, algorithm and applications. *Knowledge-Based Systems*, 207:106278, 2020.

H. Xu, C. Caramanis, and S. Mannor. Robustness and regularization of support vector machines. *Journal of Machine Learning Research*, 10(7):1485–1510, 2009.

H. Xu, C. Caramanis, and S. Mannor. Robust regression and Lasso. *IEEE Transactions on Information Theory*, 56(7):3561–3574, 2010a.

Z. Xu, R. Jin, H. Yang, I. King, and M. R. Lyu. Simple and efficient multiple kernel learning by group Lasso. In *Proceedings of the 27th International Conference on Machine Learning*, pages 1175–1182, 2010b.

H. Yang, Z. Xu, I. King, and M. R. Lyu. Online learning for group Lasso. In *Proceedings of the 27th International Conference on Machine Learning*, pages 1191–1198, 2010.

L. Yang, J. Li, D. F. Sun, and K.-C. Toh. A fast globally linearly convergent algorithm for the computation of Wasserstein barycenters. *Journal of Machine Learning Research*, 22 (21):1–37, 2021.

J. Ye, P. Wu, J. Z. Wang, and J. Li. Fast discrete distribution clustering using Wasserstein barycenter with sparse support. *IEEE Transactions on Signal Processing*, 65(9):2317–2332, 2017.

K. Yosida. *Functional Analysis*. Springer, Berlin, 1964.

M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(1):49–67, 2006.

X. Zeng and M. A. Figueiredo. Decreasing weighted sorted $\ell_1$ regularization. *IEEE Signal Processing Letters*, 21(10):1240–1244, 2014.

C.-H. Zhang and J. Huang. The sparsity and bias of the Lasso selection in high-dimensional linear regression. *The Annals of Statistics*, 36(4):1567–1594, 2008.

Y. Zhang, N. Zhang, D. F. Sun, and K.-C. Toh. An efficient Hessian based algorithm for solving large-scale sparse group Lasso problems. *Mathematical Programming*, 179(1): 223–263, 2020.