Insights and algorithms for the multivariate square-root lasso

Aaron J. Molstad
Department of Statistics
University of Florida
amolstad@ufl.edu

Abstract

We study the multivariate square-root lasso, a method for fitting the multivariate response linear regression model with dependent errors. This estimator minimizes the nuclear norm of the residual matrix plus a convex penalty. Unlike some existing methods for multivariate response linear regression, which require explicit estimates of the error covariance matrix or its inverse, the multivariate square-root lasso criterion implicitly adapts to dependent errors and is convex. To justify the use of this estimator, we establish an error bound which illustrates that like the univariate square-root lasso (Belloni et al., 2011), the multivariate square-root lasso is pivotal with respect to the unknown error covariance matrix. Based on our theory, we propose a simple tuning approach which requires fitting the model for only a single value of the tuning parameter, i.e., does not require cross-validation. We propose two algorithms to compute the estimator: a prox-linear alternating direction method of multipliers algorithm, and a fast first order algorithm which can be applied in special cases. In both simulation studies and a real data application, we show that the multivariate square-root lasso can outperform more computationally intensive methods which estimate both the regression coefficient matrix and error precision matrix.

Keywords: pivotal estimation, multivariate response linear regression, convex optimization, covariance and precision matrix estimation

1 Introduction

Multivariate response linear regression is a classical method for modeling the linear relationship between a p-variate vector of predictors and a q-variate vector of responses. In this article, we will assume that the n observed response vectors $y_1, \ldots, y_n$ are realizations of the random vectors

$$\beta_{*0} + \beta_*' x_i + \epsilon_i, \quad i = 1, \ldots, n, \tag{1}$$

where $\beta_{*0} \in \mathbb{R}^q$ and $\beta_* \in \mathbb{R}^{p\times q}$ are the unknown intercept vector and regression coefficient matrix, respectively; and the $x_i \in \mathbb{R}^p$ are the measured predictors for the ith subject. We assume that the ϵ_i's are independent and identically distributed q-variate random vectors with mean zero and covariance $\Sigma_* \in \mathbb{S}_+^q$ where Σ_* is unknown and $\mathbb{S}_+^q$ denotes the set of $q \times q$ symmetric and positive definite matrices. Let $\Omega_* \equiv \Sigma_*^{-1}$ be the unknown error precision matrix. For notational convenience, let $Y = (y_1 - \bar{y}, \ldots, y_n - \bar{y})' \in \mathbb{R}^{n\times q}$ and $X = (x_1 - \bar{x}, \ldots, x_n - \bar{x})' \in \mathbb{R}^{n\times p}$, where $\bar{y} = n^{-1}\sum_{i=1}^n y_i$ and $\bar{x} = n^{-1}\sum_{i=1}^n x_i$.

Many methods exist for fitting the multivariate response linear regression model (1). When $n > p$ and the ϵ_i's are multivariate normal, the maximum likelihood estimator (and equivalently, least squares estimator) of β_* does not require knowledge of nor an estimate of Ω_*. When $p \geq n$, in which case the least squares estimator is not unique, a popular alternative is to estimate β_* by minimizing a penalized least squares criterion (i.e., penalized squared Frobenius norm criterion) using penalties that exploit the matrix structure of the unknown regression coefficients (Turlach et al., 2005; Yuan et al., 2007; Obozinski et al., 2011). However, the penalized least squares criterion implicitly assumes $\Sigma_* \propto I_q$, e.g., the penalized least squares estimator is equivalent to the penalized normal maximum likelihood estimator when $\Sigma_* \propto I_q$ and is known.

This limitation of penalized least squares has motivated numerous methods which incorporate an estimate of Ω_* into the estimation procedure for β_*. One class of methods jointly estimate Ω_* and β_* by maximizing a penalized normal log-likelihood (Rothman et al., 2010; Yin and Li, 2011), using ℓ_1-norm penalties on both the entries of the optimization variable corresponding to β_* and off-diagonal entries of the optimization variable corresponding to Ω_*. Alternatively, Wang (2015) proposed a method which performs estimation column-by-column, estimating the kth column of β_* and Ω_* jointly for $k = 1, \ldots, q$. While these methods can perform well in certain settings, an estimate of Ω_* is often not needed by the practitioner yet requires estimating $O(q^2)$ additional parameters, increases the computational burden, and in the case of Rothman et al. (2010) and Yin and Li (2011), requires solving a non-convex optimization problem.

An ideal estimation criterion for β_* would be convex and could account for dependent errors without requiring an explicit estimate of Ω_*. To this end, we study the class of estimators

$$\hat{\beta} = \underset{\beta\in\mathbb{R}^{p\times q}}{\arg\min} \left\{ \frac{1}{\sqrt{n}}\|Y - X\beta\|_* + \lambda\mathcal{P}(\beta) \right\}, \tag{2}$$

where $\|A\|_*$ denotes the nuclear norm of a matrix A, i.e., the norm which sums the singular values of its argument, $\mathcal{P} : \mathbb{R}^{p\times q} \to \mathbb{R}_+$ is a penalty function, and λ is a positive, user specified tuning parameter. When $\mathcal{P}$ is convex, which we will assume throughout, the objective function in (2) is convex. For concreteness, we focus on (2) with $\mathcal{P}(\beta) = \sum_{j,k} |\beta_{j,k}|$, i.e., the ℓ_1-norm penalty unless stated otherwise.

Van de Geer and Stucky (2016) and Van de Geer (2016) proposed the ℓ_1-penalized version of (2), which they called the multivariate square-root lasso. Their focus was on using (2) to construct confidence sets for high-dimensional regression coefficient vectors in univariate response linear regression. Here, we focus on (2) as a method for fitting (1) in high-dimensional settings.

Computing (2) is non-trivial: the nuclear norm, though convex, is non-differentiable and thus (2) is the sum of two non-differentiable functions. To date, there are no existing specialized algorithms to compute (2) with convergence guarantees. Van de Geer and Stucky (2016) suggested a fixed-point iterative procedure for computing (2), but its unclear whether the iterates will converge to a KKT point for non-trivial values of the tuning parameter (see Section 3.3). In a later version of Van de Geer and Stucky (2016) appearing in a PhD thesis, Stucky (2017) computed (2) using the general purpose convex solver `CVX` (Grant and Boyd, 2014), which can be slow in high-dimensional settings.

In addition to the computational challenges, (2) has not been studied in terms of its finite-sample or asymptotic properties. While Van de Geer and Stucky (2016) and Van de Geer (2016) pointed out the connection between (2) and the univariate square-root lasso (Belloni et al., 2011), their focus was on (2) as a means for constructing confidence intervals. They did not study the statistical properties of (2) nor did they explore the empirical performance of (2) in the context of fitting multivariate response linear regression models.

In this article, we study (2) from theoretical, computational, and empirical perspectives. In particular, we prove that, like the univariate square-root lasso, (2) is pivotal in the sense that the tuning parameter leading to near-oracle performance does not depend on the unknown error covariance Σ_*. In so doing, we establish an error bound for the ℓ_1-penalized version of (2). We argue that (2), like the univariate square-root lasso, has a self-normalizing property in that it implicitly incorporates an estimator of the error precision matrix into the criterion for estimating β_*. Through simulation studies, we show that (2) can perform as well or better than methods which estimate both β_* and Ω_* jointly, both of which outperform penalized least squares estimators when Σ_* has many nonzero off-diagonals. Based on our theory, we also propose a tuning procedure which does not require cross-validation: it requires solving (2) for only a single value of the tuning parameter. Finally, we propose two new algorithms to compute (2) efficiently: one algorithm which can be used in any setting and has convergence guarantees, and a second which can be much faster when $n > q$ and the tuning parameter is sufficiently large. An R package implementing our method is available for download at github.com/ajmolstad/MSRL.

Throughout, when we write $(U, D, V) = \text{svd}(A)$, we refer to the singular value decomposition of $A = UDV' \in \mathbb{R}^{a\times b}$ where, letting $s = \min\{a, b\}$, $U \in \mathbb{R}^{a\times s}$, $D \in \mathbb{R}^{s\times s}$ and $V \in \mathbb{R}^{b\times s}$ where D is diagonal with nonnegative entries and U and V are semi-orthogonal. Define the norms $\|A\|_F^2 = \sum_{j,k} A_{j,k}^2$; $\|A\|_{\max} = \max_{j,k} |A_{j,k}|$ and $\|A\|_2 = \sigma_1(A)$ where $\sigma_j(A)$ denotes the jth largest singular value of A. We will refer to the norm $\|\cdot\|_{\max}$ as the "max-norm".

2 The multivariate square-root lasso

2.1 Review of the univariate square-root lasso

The univariate square-root lasso (Belloni et al., 2011) is a method for fitting high-dimensional univariate response (i.e., $q = 1$) linear regression models. Momentarily, suppose we are interested in fitting a univariate

response linear regression model where, given measured predictors $x_i \in \mathbb{R}^p$ for $i = 1, \ldots, n$, we assume the measured responses $z_1, \ldots, z_n$ are realizations of

$$\gamma_{*0} + \gamma_*' x_i + u_i, \quad i = 1, \ldots, n,$$

where $\gamma_{*0} \in \mathbb{R}$ is the unknown intercept, $\gamma_* \in \mathbb{R}^p$ is the unknown regression coefficient vector, and the $u_i \in \mathbb{R}$ are independent with mean zero and variance σ_*^2. Let $Z = (z_1 - \bar{z}, \ldots, z_n - \bar{z})' \in \mathbb{R}^n$ with $\bar{z} = n^{-1} \sum_{i=1}^n z_i$, and let $u = (u_1, \ldots, u_n)' \in \mathbb{R}^n$ be the unobserved vector of errors.

A well known property of the ℓ_1-penalized least squares estimator of γ_*, as it's defined in Bickel et al. (2009), is that the tuning parameter leading to optimal performance depends on the error variance σ_*^2 (Bickel et al., 2009). In particular, for the ℓ_1-penalized least squares estimator with tuning parameter λ^{lasso}, the tuning parameter which provides near-oracle convergence rates is the smallest λ^{lasso} which is greater than or equal to c times the max-norm of the score vector evaluated at γ_* with high probability:

$$\lambda^{\text{lasso}} \geq \frac{c}{n} \|X'u\|_{\max} = \sigma_* \frac{c}{n} \|X'v\|_{\max} \tag{3}$$

where $c > 1$ is some constant (set to 1.1 in Belloni et al. (2011)); and $v \in \mathbb{R}^n$ has entries with mean zero and unit variance. Of course, (3) cannot be used in practice as it depends on the unknown quantity σ_*. Instead practitioners use cross-validation or an information criterion to select the tuning parameter which requires fitting the model many times over a set of candidate tuning parameters.

As an alternative, Belloni et al. (2011) proposed the univariate square-root lasso estimator

$$\hat{\gamma}^{\sqrt{\text{lasso}}} = \underset{\gamma \in \mathbb{R}^p}{\arg\min} \left\{ \frac{1}{\sqrt{n}} \|Z - W\gamma\|_2 + \lambda^{\sqrt{\text{lasso}}} \sum_{j=1}^p |\gamma_j| \right\}.$$

The univariate square-root lasso can be interpreted as method which simultaneously estimates both the variance and regression coefficient vector since

$$(\hat{\gamma}^{\sqrt{\text{lasso}}}, \hat{\sigma}) = \underset{\gamma \in \mathbb{R}^p, \sigma > 0}{\arg\min} \left\{ \frac{\|Z - W\gamma\|_2^2}{2n\sigma} + \frac{\sigma}{2} + \lambda^{\sqrt{\text{lasso}}} \sum_{j=1}^p |\gamma_j| \right\}. \tag{4}$$

where $\sqrt{n}\hat{\sigma} = \|Z - W\hat{\gamma}^{\sqrt{\text{lasso}}}\|_2$, provided that $\|Z - W\hat{\gamma}^{\sqrt{\text{lasso}}}\|_2 \neq 0$. That is, the square-root lasso self-normalizes by scaling the residual sum of squares by an estimate of the inverse standard deviation. The alternative formulation in (4), called the scaled lasso, was studied by Sun and Zhang (2012). Similar to ℓ_1-penalized least squares, if $\lambda^{\sqrt{\text{lasso}}}$ is greater than or equal to c times the max-norm of the score vector evaluated at γ_* with high probability, e.g.,

$$\lambda^{\sqrt{\text{lasso}}} \geq \frac{c}{\sqrt{n}} \frac{\|X'u\|_{\max}}{\|u\|_2} = \frac{c}{\sqrt{n}} \|X'h\|_{\max}, \quad h = u/\|u\|_2 \in \mathbb{R}^n, \tag{5}$$

then $\hat{\gamma}^{\sqrt{\text{lasso}}}$ achieves the same rate of convergence as the ℓ_1-penalized least squares estimator with the tuning parameter in (3) (Belloni et al., 2011). Remarkably, the lower bound in (5) does not depend on any unknown parameters since the distribution of h does not depend on σ_*, so in principle $\lambda^{\sqrt{\text{lasso}}}$ can be chosen based on quantiles of $\|X'h\|_{\max}$.

There exists a method which extends the univariate square-root lasso framework to the multivariate response linear regression setting. Liu et al. (2015) proposed an estimator which minimizes the sum of the euclidean norm of residuals for each response plus a nuclear norm or group-lasso penalty on the optimization variable corresponding to β_*. However, the method of Liu et al. (2015) assumes that Σ_* is diagonal (although not necessarily proportional to I_q). In addition, when the penalty is separable across the columns of its matrix argument, the method of Liu et al. (2015) is equivalent to performing q separate univariate square-root lasso regressions. For more details, see our description of their method in Section 5.2.

2.2 Implicit covariance estimation

In this and the following subsection, we demonstrate that (2) generalizes the univariate square-root lasso to the multivariate response linear regression setting. When $q = 1$ it is trivial to show that (2) is equivalent to the univariate square-root lasso. Moreover, like the univariate square-root lasso, the multivariate square-root lasso also has a self-normalizing interpretation because

$$\frac{1}{\sqrt{n}}\|Y - X\beta\|_* = \frac{1}{n}\text{tr}\left\{(Y - X\beta)\tilde{\Sigma}_\beta^-(Y - X\beta)'\right\}, \text{ where } \tilde{\Sigma}_\beta = \frac{1}{\sqrt{n}}[(Y - X\beta)'(Y - X\beta)]^{\frac{1}{2}},$$

and A^- denotes the Moore-Penrose psuedoinverse of A. That is, the nuclear norm of residuals can be expressed as a weighted residual sum of squares where the weight is an estimate the square-root error precision matrix Ω_*. In fact, the multivariate square-root lasso can also be interpreted as jointly estimating the error covariance and regression coefficient matrix.

Remark 1 *(Van de Geer, 2016) Let*

$$(\bar{\beta}, \bar{\Sigma}) = \underset{\beta \in \mathbb{R}^{p\times q}, \Sigma \succ 0}{\arg\min} \left\{ \frac{1}{2n}\text{tr}\left[(Y - X\beta)\Sigma^{-\frac{1}{2}}(Y - X\beta)'\right] + \frac{\text{tr}(\Sigma^{\frac{1}{2}})}{2} + \lambda\mathcal{P}(\beta) \right\}. \tag{6}$$

If $Y - X\hat{\beta}$ has q nonzero singular values, then the estimator in (6) *satisfies $\bar{\Sigma} = \frac{1}{n}(Y - X\hat{\beta})'(Y - X\hat{\beta})$ and $\bar{\beta} = \hat{\beta}$.*

Remark 1 suggests that we can solve the joint convex optimization problem (6) by solving (2) – we need not explicitly estimate Σ_* or its inverse to account for dependent errors. In the simulation studies, we demonstrate that this implicit covariance estimation provides estimates of β_* which perform similarly to estimators which estimate or use the true value of Ω_* in their estimation criterion.

2.3 Statistical properties of the multivariate square-root lasso

We establish an error bound for $\hat{\beta} - \beta_*$ which allows p and q to grow with n. In obtaining this bound we prove that (2), like the univariate square-root lasso, is pivotal with respect to Ω_*. For concreteness, we focus on (2) with $\mathcal{P}(\beta) = \sum_{j,k} |\beta_{j,k}|$. We assume that the regression coefficient matrix β_* has $s \geq 1$ nonzero entries whose indices are denoted by the set $\mathcal{S}$, i.e., $\mathcal{S} = \{(j,k) : \beta_{*j,k} \neq 0\}$. In addition, we treat X as fixed and assume its columns have been normalized so that $\|X_j\|_2 = 1$ for $j = 1, \ldots, p$. Define $\bar{c} = (c+1)/(c-1)$. We will require the following assumptions:

A1. The $n \times q$ error matrix $\mathcal{E} = Y - X\beta_*$ has q nonzero singular values almost surely.

A2. The distribution of the error matrix $\mathcal{E}$ is left-spherical, i.e., for any orthogonal matrix $O_n^{(n)} \in \mathbb{R}^{n \times n}$, $O_n^{(n)}\mathcal{E}$ has the same matrix-variate distribution as $\mathcal{E}$.

Assumption A1 requires that $n > q$ and that the q-variate distribution of each row of $\mathcal{E}$ is non-degenerate. Assumptions A1 and A2 would hold if, for example, the rows of $\mathcal{E}$ were independent and each row followed a mean zero, q-variate multivariate normal distribution with covariance $\Sigma_* \in \mathbb{S}_+^q$. In addition to A1 and A2, our bounds will depend on the quantity:

$$\kappa_{\mathcal{E},c} = \inf_{\Delta \in \mathcal{C}(\mathcal{S},c)} \left\{ \frac{\sup_{\|Q\| \leq 1} \operatorname{tr}\left\{(Q - U_* V_*')'(U_* D_* V_*' - X\Delta)\right\}}{\sqrt{n}\|\Delta\|_F^2} \right\},$$

where $\mathcal{C}(\mathcal{S}, c) = \left\{ \Delta \in \mathbb{R}^{p \times q} : \Delta \neq 0, \bar{c} \sum_{(j,k) \notin \mathcal{S}} |\Delta_{j,k}| \leq \sum_{(j,k) \in \mathcal{S}} |\Delta_{j,k}| \right\}$ and $(U_*, D_*, V_*) = \operatorname{svd}(\mathcal{E})$. The quantity $\kappa_{\mathcal{E},c}$ is needed to establish a lower bound on the first order Taylor expansion of the nuclear norm of the residuals. A positive $\kappa_{\mathcal{E},c}$ exists almost surely since the numerator is positive with probability one under A1. Unlike in the least squares setting, this quantity is not a restricted eigenvalue: $\kappa_{\mathcal{E},c}$ also depends on the errors $Y - X\beta_*$. However, if the regression were error-less, $\kappa_{\mathcal{E},c}$ would simplify to a restricted singular value-type quantity: $\inf_{\Delta \in \mathcal{C}(\mathcal{S},c)} \left\{ \|X\Delta\|_* / \sqrt{n}\|\Delta\|_F^2 \right\}$. Using that $\|A\|_* = \sup_{\|W\| \leq 1} \operatorname{tr}(W'A)$, it's immediate that the Q which maximizes the numerator is $Q = \tilde{U}\tilde{V}'$, where $(\tilde{U}, \tilde{D}, \tilde{V}) = \operatorname{svd}(\mathcal{E} - X\Delta)$. Thus, if the entries of $X\Delta$ are large relative to the entries of $\mathcal{E}$, then $\kappa_{\mathcal{E},c}$ will be large. Conversely, if the error variances and off-diagonals of Σ_* are large, $\kappa_{\mathcal{E},c}$ will be small because $\mathcal{E} - X\Delta \approx U_* D_* V_*'$ since the diagonals of D_* would be large. Assigning a probability to a particular value of $\kappa_{\mathcal{E},c}$ is technically difficult because we allow the columns of $Y - X\beta_*$ to be dependent. Thus, we do not make any assumptions about $\kappa_{\mathcal{E},c}$ – this quantity appears in the error bounds we establish below.

Proposition 1 *Assume A1 is true. Let* $(U_*, D_*, V_*) = \operatorname{svd}(Y - X\beta_*)$. *For any constant* $c > 1$, *if* $c\|X'U_*V_*'\|_{\max} \leq \sqrt{n}\lambda$, *then* $\|\hat{\beta} - \beta_*\|_F \leq \frac{\bar{c}}{\kappa_{\mathcal{E},c}} \lambda \sqrt{s}$. *If A2 is also true, then the distribution of* $X'U_*V_*'$ *does not depend on* Ω_*.

The proof of Proposition 1 can be found in the Supplementary Material. Proposition 1 reveals that the error $\|\hat{\beta} - \beta_*\|_F$ is controlled by the max-norm of the score, i.e., the quantity $\frac{1}{\sqrt{n}}\|X'U_*V_*'\|_{\max}$ (see proof of

Lemma **??** in the Supplementary Material for justification of this score); the number of nonzero entries in β_*, s; and the quantity $\kappa_{\mathcal{E},c}$, through which the error covariance Σ_* affects the error bound. Similar to the univariate square-root lasso, the tuning parameter leading to the smallest error bound is the minimum λ which is greater than or equal to $\frac{c}{\sqrt{n}}\|X'U_*V_*'\|_{\max}$ with high probability. Under A1 and A2, U_*V_*' is a random matrix uniformly distributed on the set of $n \times q$ semi-orthogonal matrices $O_q^{(n)}$ such that ${O_q^{(n)}}'O_q^{(n)} = I_q$ (Eaton, 1989). Hence, Proposition 1 verifies that the multivariate square-root lasso is pivotal with respect to Ω_*.

The result of Proposition 1 also suggests that the tuning parameter λ could be selected via Monte-Carlo approximation of quantiles of the distribution of $\|X'O_q^{(n)}\|_{\max}$ where $O_q^{(n)} \in \mathbb{R}^{n\times q}$ is a random matrix which is uniformly distributed on the set of $n \times q$ semi-orthogonal matrices. For example, the result of Proposition 1 would hold with probability $1-\alpha$ if we selected λ equal to the $(1-\alpha)$th quantile of the distribution of $\frac{c}{\sqrt{n}}\|X'O_q^{(n)}\|_{\max}$ for some $c > 1$. We explore this approach in Section 4.4.

Finally, we can use the distribution of U_*V_*' to establish an explicit choice of λ which yields a more insightful asymptotic error bound.

Theorem 1 *Assume A1 and A2 are true. Let $\tilde{c}$, c, α, and ϕ be fixed constants such that $\tilde{c} > c > 1$, $\alpha \in (0,1)$, and $\phi > 1$. Let $\dot{c} = \tilde{c}/c$ and $\hat{c} = [\tilde{c}(c+1)]/(c-1)$. If $\lambda = \tilde{c}\left\{2n^{-1}\log(2\phi pq/\alpha)\right\}^{1/2}$ and $n(\dot{c}^2-1)^2 \geq 4\dot{c}^4\log[\{\alpha(\phi-1)\}^{-1}\phi q]$, then*

$$\|\hat{\beta} - \beta_*\|_F \leq \frac{\hat{c}}{\kappa_{\mathcal{E},c}}\sqrt{\frac{2s\log(2\phi pq/\alpha)}{n}}$$

with probability at least $1-\alpha$.

We prove Theorem 1 in the Supplementary Material. Theorem 1 reveals that for sufficiently large n, we can set λ equal to an explicit quantity which will satisfy the condition of Proposition 1 with high probability. We also explore this choice of tuning parameter in Section 4.4. Importantly, Theorem 1 suggests that (2) will perform well in high dimensions since both p and q scale the error bound logarithmically.

The quantity ϕ in Theorem 1 illustrates a tradeoff between the sample size and the effect of p and q on the error bound. To obtain the smallest error bound, one would take ϕ as small as possible such that $\phi > 1$ and $n(\dot{c}^2-1)^2 \geq 4\dot{c}^4\log[\{\alpha(\phi-1)\}^{-1}\phi q]$. Hence, a larger sample size would allow for smaller ϕ, and thus, a reduced effect of p and q.

3 Computation

3.1 Properties of the solution

In the low-dimensional setting $\max(p,q) < n$, the minimizer of the unpenalized nuclear norm of residuals is equivalent to the minimizer of the unpenalized squared Frobenius norm of residuals. That is, the least squares estimator $(X'X)^{-1}X'Y$, when it exists, is a minimizer of (2) when $\lambda = 0$. The penalized solution,

however, does not coincide with the penalized least squares estimator. This can be seen by examining the first order conditions for (2) which we characterize in the following remark.

Remark 2 *When $Y - X\hat{\beta}$ has q nonzero singular values, the first order conditions for* (2)*, which are necessary and sufficient for optimality, are*

$$\frac{1}{\sqrt{n}} X'(Y - X\hat{\beta})[(Y - X\hat{\beta})'(Y - X\hat{\beta})]^{-\frac{1}{2}} \in \lambda \partial \mathcal{P}(\hat{\beta}) \tag{7}$$

where $\partial \mathcal{P}(\hat{\beta})$ is the subgradient of $\mathcal{P}$ evaluated at $\hat{\beta}$. The left hand side of condition (7) *could instead be written in terms of the singular vectors of $Y - X\hat{\beta}$ since $X'(Y - X\hat{\beta})[(Y - X\hat{\beta})'(Y - X\hat{\beta})]^{-\frac{1}{2}} = X'U_{\hat{\beta}}V_{\hat{\beta}}'$ where $(U_{\hat{\beta}}, D_{\hat{\beta}}, V_{\hat{\beta}}) = \mathrm{svd}(Y - X\hat{\beta})$. If $Y - X\hat{\beta}$ has fewer than q nonzero singular values, the first order conditions for* (2) *are $0 \in -\frac{1}{\sqrt{n}} X'(U_{\hat{\beta}}V_{\hat{\beta}}' + W_{\hat{\beta}}) + \lambda \partial \mathcal{P}(\hat{\beta})$, where $W_{\hat{\beta}} \in \{W \in \mathbb{R}^{n \times q} : \|W\| \leq 1, U_{\hat{\beta}}'W = 0, WV_{\hat{\beta}} = 0, (U_{\hat{\beta}}, D_{\hat{\beta}}, V_{\hat{\beta}}) = \mathrm{svd}(Y - X\hat{\beta})\}$.*

See the proof of Lemma **??** of the Supplementary Material for the derivation of Remark 2. Of course, $Y - X\hat{\beta}$ can only have q nonzero singular values when $n > q$ and when λ is sufficiently large. In these cases, we use (7) as a termination criterion – see Section 3.3.

In the following subsections, we propose two algorithms to compute (2) for general penalty functions $\mathcal{P}$ which are convex and coercive. The first can be used in any setting, and the second will be applicable in the case that $n > q$ and λ is sufficiently large, but can be much faster.

3.2 Prox-linear ADMM

To compute (2), we must address that neither the penalty nor the nuclear norm of residuals are differentiable in general. We use a variation of the alternating direction method of multipliers (ADMM) algorithm which allows us to operate on the nuclear norm and penalty separately through their proximal operators (Parikh and Boyd, 2014). Throughout this and the subsequent section, we will refer to a proximal operator of a function f, which is defined as

$$\mathrm{Prox}_f(y) = \arg\min_x \left\{ \frac{1}{2}\|y - x\|_F^2 + f(x) \right\}.$$

For many popular penalty functions, the proximal operator can be computed efficiently. For closed-form solutions to the proximal operators of penalty functions often used in multivariate response linear regression, including the nuclear norm, see Chapter 6 of Parikh and Boyd (2014).

Following Boyd et al. (2011), we first introduce an additional primal variable $\Phi \in \mathbb{R}^{n \times q}$, so that we can rewrite (2) as the constrained optimization problem:

$$\arg\min_{\beta \in \mathbb{R}^{p \times q}, \Phi \in \mathbb{R}^{n \times q}} \left\{ \|\Phi\|_* + \tilde{\lambda}\mathcal{P}(\beta) \right\}, \quad \Phi = Y - X\beta, \tag{8}$$

where $\tilde{\lambda} = \sqrt{n}\lambda$. To apply the ADMM algorithm, we operate on the augmented Lagrangian for the constrained problem in (8)

$$\mathcal{G}_\rho(\beta, \Phi, \Gamma) = \|\Phi\|_* + \tilde{\lambda}\mathcal{P}(\beta) + \text{tr}\left\{\Gamma'(Y - X\beta - \Phi)\right\} + \frac{\rho}{2}\|Y - X\beta - \Phi\|_F^2,$$

where $\rho > 0$ is a step size parameter and $\Gamma \in \mathbb{R}^{n\times q}$ is the Lagrangian dual variable. Then, the updating equations for the $(k+1)$th iterate of the standard ADMM algorithm are

$$\Phi^{(k+1)} = \underset{\Phi\in\mathbb{R}^{n\times q}}{\arg\min}\, \mathcal{G}_\rho(\beta^{(k)}, \Phi, \Gamma^{(k)}) \tag{9}$$

$$\beta^{(k+1)} = \underset{\beta\in\mathbb{R}^{p\times q}}{\arg\min}\, \mathcal{G}_\rho(\beta, \Phi^{(k+1)}, \Gamma^{(k)}) \tag{10}$$

$$\Gamma^{(k+1)} = \Gamma^{(k)} + \tau\rho(Y - X\beta^{(k+1)} - \Phi^{(k+1)}), \tag{11}$$

where τ is a parameter which rescales the step size for the dual variable update. Previous work has shown that $\tau \in (0, \frac{1+\sqrt{5}}{2})$ can lead to convergence of ADMM under standard conditions (Deng and Yin, 2016).

The first updating equation of the ADMM algorithm, (9), can be expressed in terms of the proximal operator of the nuclear norm:

$$\Phi^{(k+1)} = \text{Prox}_{\rho^{-1}\|\cdot\|_*}\left(Y + \rho^{-1}\Gamma^{(k)} - X\beta^{(k)}\right)$$

which can be solved efficiently in closed form by computing the singular value decomposition of $Y + \rho^{-1}\Gamma^{(k)} - X\beta^{(k)}$ and soft thresholding its singular values (see Steps 1 and 2 of Algorithm 1).

When p is large, the second step of the ADMM algorithm, (10), is more computationally burdensome since it involves solving a penalized least squares problem:

$$\underset{\beta\in\mathbb{R}^{p\times q}}{\arg\min}\, \mathcal{G}_\rho(\beta, \Phi^{(k+1)}, \Gamma^{(k)}) = \underset{\beta\in\mathbb{R}^{p\times q}}{\arg\min}\left\{\frac{1}{2}\|Y + \rho^{-1}\Gamma^{(k)} - \Phi^{(k+1)} - X\beta\|_F^2 + \frac{\tilde{\lambda}}{\rho}\mathcal{P}(\beta)\right\}. \tag{12}$$

To avoid solving (12) at every iteration, we instead minimize a majorizing function of $\mathcal{G}_\rho(\beta, \Phi^{(k+1)}, \Gamma^{(k)})$ constructed at the previous iterate $\beta^{(k)}$. Specifically, we majorize $\mathcal{G}_\rho(\beta, \Phi^{(k+1)}, \Gamma^{(k)})$ in (10) with

$$\mathcal{M}_{\rho,\eta}(\beta, \Phi^{(k+1)}, \Gamma^{(k)}; \beta^{(k)}) \equiv \mathcal{G}_\rho(\beta, \Phi^{(k+1)}, \Gamma^{(k)}) + \frac{\rho}{2}\text{tr}\{(\beta - \beta^{(k)})'Q_\eta(\beta - \beta^{(k)})\},$$

where $Q_\eta = \eta I_p - X'X$ with constant $\eta \in \mathbb{R}$ chosen so that $\eta I_p \succeq X'X$. Then, we replace (10) with

$$\begin{aligned}\beta^{(k+1)} &= \underset{\beta\in\mathbb{R}^{p\times q}}{\arg\min}\, \mathcal{M}_\rho(\beta, \Phi^{(k+1)}, \Gamma^{(k)}; \beta^{(k)}) \\ &= \text{Prox}_{(\rho\eta)^{-1}\tilde{\lambda}\mathcal{P}}\left\{\beta^{(k)} + \eta^{-1}X'\left(Y + \rho^{-1}\Gamma^{(k)} - \Phi^{(k+1)} - X\beta^{(k)}\right)\right\}.\end{aligned} \tag{13}$$

It follows that using (13), $\mathcal{G}_\rho(\beta^{(k+1)}, \Phi^{(k+1)}, \Gamma^{(k)}) \leq \mathcal{G}_\rho(\beta^{(k)}, \Phi^{(k+1)}, \Gamma^{(k)})$ by the majorize-minimize

Algorithm 1 Prox-linear ADMM for (2)

Input: Initialize $\rho > 0$, $\eta > \varphi_1(X'X)$, $\tilde{\lambda} = \sqrt{n}\lambda$, $\tau \in (0, \frac{1+\sqrt{5}}{2})$, $r = \min(p, q)$and $k = 0$.

1: Decompose $(U, D, V) = \text{svd}(Y + \rho^{-1}\Gamma^{(k)} - X\beta^{(k)})$.

2: Compute $\Phi^{(k+1)} \leftarrow U\{\texttt{pmax}(D - \rho^{-1}I_r, 0)\}V'$

3: Compute $\beta^{(k+1)} \leftarrow \text{Prox}_{(\rho\eta)^{-1}\tilde{\lambda}\mathcal{P}}\left\{\beta^{(k)} + \eta^{-1}X'\left(Y + \rho^{-1}\Gamma^{(k)} - \Phi^{(k+1)} - X\beta^{(k)}\right)\right\}$

4: Compute $\Gamma^{(k+1)} \leftarrow \Gamma^{(k)} + \tau\rho(Y - X\beta^{(k+1)} - \Phi^{(k+1)})$

5: If not converged, set $k \leftarrow k + 1$ and return to 1.

principle (Lange, 2016). In the case that $\mathcal{P}$ is the ℓ_1-norm, (13) is computed by soft-thresholding $\beta^{(k)} + \eta^{-1}X'(Y + \rho^{-1}\Gamma^{(k)} - \Phi^{(k+1)} - X\beta^{(k)})$. A complete derivation of the $\beta^{(k+1)}$ update is provided in the Supplementary Material and the full algorithm is summarized in Algorithm 1.

This variation of the alternating direction method of multipliers algorithm is called the prox-linear alternating direction method of multipliers. A similar approach was used for computing penalized quantile regression estimators by Gu et al. (2018). Fortunately, this approximation scheme maintains the convergence properties of the ADMM algorithm – applying the result of Theorem 2.2 from Deng and Yin (2016), we have the following convergence guarantee:

Proposition 2 *(Deng and Yin, 2016) Suppose* $0 < \tau < \frac{1+\sqrt{5}}{2}$*,* $\rho > 0$*, and* $\eta \geq \varphi_1(X'X)$ *are fixed. Then, for iterates* $(\Phi^{(k)}, \beta^{(k)}, \Gamma^{(k)})$ *generated from Algorithm 1, it follows that* $\Phi^{(k)} \to \Phi^{\dagger}$*,* $\beta^{(k)} \to \beta^{\dagger}$ *and* $\Gamma^{(k)} \to \Gamma^{\dagger}$ *as* $k \to \infty$ *where* $(\Phi^{\dagger}, \beta^{\dagger}, \Gamma^{\dagger})$ *satisfy the KKT conditions for* (8)*.*

The proof of Proposition 3 is omitted as it follows directly from the arguments of Deng and Yin (2016). The termination conditions we use are based on the dual and primal residuals suggested by Boyd et al. (2011).

3.3 Fast first order algorithm for large n

The nuclear norm is non-differentiable in general. The subgradient of the nuclear norm of the residual matrix with respect to β is:

$$\partial\|Y - X\beta\|_* = \left\{-X'U_\beta V_\beta' - X'W : \|W\| \leq 1, U_\beta' W = WV_\beta = 0, (U_\beta, D_\beta, V_\beta) = \text{svd}(Y - X\beta)\right\},$$

for example, see Watson (1992). However, when $Y - X\beta$ has q non-zero singular values, the subgradient of $\|Y - X\beta\|_*$ with respect to β is a matrix in $\mathbb{R}^{p\times q}$:

$$\nabla\|Y - X\beta\|_* = -X'(Y - X\beta)\left[(Y - X\beta)'(Y - X\beta)\right]^{-\frac{1}{2}} = -X'U_\beta V_\beta', \tag{14}$$

so that $\|Y - X\beta\|_*$ can effectively be treated as differentiable in this situation.

This simple fact suggests that in settings where $n > q$ and λ is sufficiently large, we can use first order algorithms to solve (2). Namely, given some iterate $\beta^{(k)}$, we construct a function which majorizes

the objective function from (2) at $\beta^{(k)}$, and obtain our next iterate by minimizing that majorizing function. Letting $U_{\beta^{(k)}}$ and $V_{\beta^{(k)}}$ denote the left and right singular vectors of $Y - X\beta^{(k)}$ respectively, it follows that

$$\frac{1}{\sqrt{n}}\|Y - X\beta\|_* + \lambda\mathcal{P}(\beta) \leq \frac{1}{\sqrt{n}}\|Y - X\beta^{(k)}\|_* - \frac{1}{\sqrt{n}}\text{tr}\left\{V_{\beta^{(k)}}U'_{\beta^{(k)}}X(\beta - \beta^{(k)})\right\} + \frac{1}{2t_k}\|\beta - \beta^{(k)}\|_F^2 + \lambda\mathcal{P}(\beta), \tag{15}$$

for some sufficiently small t_k; and for all $\beta \in \mathcal{D}_\kappa$, $\beta^{(k)} \in \mathcal{D}_\kappa$ where $\mathcal{D}_\kappa = \left\{\beta \in \mathbb{R}^{p\times q} : \kappa^{-1} \geq \varphi_1(Y - X\beta) \geq \varphi_q(Y - X\beta) \geq \kappa\right\}$ for some $\kappa > 0$. We can then compute the $(k+1)$th iterate as the value of β which minimizes the right hand side of (15) constructed at the previous iterate

$$\beta^{(k+1)} = \underset{\beta\in\mathbb{R}^{p\times q}}{\arg\min}\left\{\frac{1}{\sqrt{n}}\text{tr}\left[V_{\beta^{(k)}}U'_{\beta^{(k)}}X(\beta^{(k)} - \beta)\right] + \frac{1}{2t_k}\|\beta^{(k)} - \beta\|_F^2 + \lambda\mathcal{P}(\beta)\right\}. \tag{16}$$

In the more familiar proximal operator notation, we can express (16) as

$$\beta^{(k+1)} = \text{Prox}_{t_k\lambda\mathcal{P}}\left(\frac{t_k}{\sqrt{n}}X'U_{\beta^{(k)}}V'_{\beta^{(k)}} + \beta^{(k)}\right)$$

This idea was also used by Li et al. (2016), who proposed a new class of first and second order algorithms for the univariate square-root lasso optimization problem. Our proposed iterative procedure is known as a proximal gradient descent algorithm (Parikh and Boyd, 2014). In our implementation, we use a variation of the monotone accelerated proximal gradient descent proposed by Beck and Teboulle (2009). We provide our specific version of the algorithm in Algorithm 2.

We found that for large λ, Algorithm 2 converged quickly, whereas for λ such that $Y - X\hat{\beta}$ has any small singular values (e.g., $< 10^{-3}$) Algorithm 1 was often faster. If at any point in our implementation of Algorithm 2, the matrices $\bar{D}$ or $\tilde{D}$ have singular values less than 10^{-3}, we switch to using Algorithm 1 and do so for all subsequent smaller values of λ.

We attempted to compare our algorithms to the iterative procedure proposed by Van de Geer and Stucky (2016). However, we found that while their procedure could decrease the objective function across iterations, the iterates at convergence did not satisfy the KKT conditions in the settings we tried.

We terminate Algorithm 2 when both $\varphi_q(Y - X\beta^{(k+1)}) > 0$ and the first order conditions from (7) are satisfied by $\beta^{(k+1)}$.

3.4 Computational complexity and tuning parameter selection

One important feature of (2) is that the computational complexity of both Algorithm 1 and Algorithm 2 are linear in p. The prox-linear ADMM algorithm requires $O(\min\{nq^2, n^2q\} + npq)$ operations per iteration, whereas Algorithm 2 (assuming $n > q$), requires $O(nq^2 + npq)$ operations per iteration. Unlike Algorithm 1 however, Algorithm 2 can be sped up dramatically by employing a stochastic approximation of the gradient.

Algorithm 2 Monotone accelerated proximal gradient descent for (2)

Input: Initialize $\beta^{(0)} \in \mathbb{R}^{p\times q}$, $\beta^{(-1)} = \beta^{(0)}$ $t_0 = 1$, $\alpha^{(0)} = 1$, $\alpha^{(-1)} = 1$, $\gamma \in (0,1)$, and $(\hat{U}, \hat{D}, \hat{V}) = \mathrm{svd}(Y - X\beta^{(0)})$. Set $k = 0$ and go to Step 1.

1: Compute $\Gamma^{(k)} \leftarrow \beta^{(k)} + \left(\frac{\alpha^{(k-1)}-1}{\alpha^{(k)}}\right)\left(\beta^{(k)} - \beta^{(k-1)}\right)$.

2: Decompose $(\tilde{U}, \tilde{D}, \tilde{V}) = \mathrm{svd}(Y - X\Gamma^{(k)})$.

3: Compute $\tilde{\beta} \leftarrow \mathrm{Prox}_{\frac{t_k\lambda}{\tau}\mathcal{P}}\left[\Gamma^{(k)} + \frac{t_k}{\sqrt{n}}X'\tilde{U}\tilde{V}'\right]$.

4: Decompose $(\bar{U}, \bar{D}, \bar{V}) = \mathrm{svd}(Y - X\tilde{\beta})$.

5: If $\mathrm{tr}(\bar{D}) < \mathrm{tr}(\tilde{D}) + \mathrm{tr}\left[\tilde{V}\tilde{U}'X(\Gamma^{(k)} - \tilde{\beta})\right] + \frac{\sqrt{n}}{2t_k}\|\Gamma^{(k)} - \tilde{\beta}\|_F^2$, go to Step 6. Else, set $t_k = \gamma t_k$ and return to Step 3.

6: If $\mathrm{tr}(\bar{D}) + \sqrt{n}\lambda\mathcal{P}(\tilde{\beta}) \leq \mathrm{tr}(\hat{D}) + \sqrt{n}\lambda\mathcal{P}(\beta^{(k)})$, set $\beta^{(k+1)} \leftarrow \tilde{\beta}$ and $\hat{D} \leftarrow \bar{D}$. Else, set $\beta^{(k+1)} \leftarrow \beta^{(k)}$.

7: Set $\alpha^{(k+1)} \leftarrow (1 + \sqrt{1 + 4\alpha^{2(k)}})/2$.

8: Set $t_{k+1} \leftarrow t_0$, $k = k + 1$, and return to Step 1.

This fast stochastic procedure is described in the Supplementary Material.

In Section 4.4, we discuss approaches for tuning parameter selection based on our theoretical results. We also found that selecting the tuning parameter λ by minimizing prediction error in a validation set or through K-fold cross validation could perform well in terms of prediction accuracy. In our default implementation, we use cross-validation and automatically prescribe a set of candidate tuning parameters based on the following remark.

Remark 3 *Suppose $n > q$ and let $(U_Y, D_Y, V_Y) = \mathrm{svd}(Y)$. If (i) $\mathcal{P}(\beta) = \sum_{j,k}|\beta_{j,k}|$ and $\lambda > \frac{1}{\sqrt{n}}\|X'U_YV_Y'\|_{\max}$; (ii) $\mathcal{P}(\beta) = \sum_j(\sum_k \beta_{j,k}^2)^{1/2}$ and $\lambda > \frac{1}{\sqrt{n}}\max_j \|[X'U_YV_Y']_{j,\cdot}\|_2$; or (iii) $\mathcal{P}(\beta) = \|\beta\|_*$ and $\lambda > \frac{1}{\sqrt{n}}\|X'U_YV_Y\|$; then $\hat{\beta} = 0$.*

Thus, we set $\lambda_{\max}$ equal to the upper bound established in Remark 3 and set $\lambda_{\min} = \delta\lambda_{\max}$ where $\delta < 1$ is some constant, e.g., we set $\delta = 0.1$ in simulations. We then consider tuning parameters from $\lambda_{\max}$ to $\lambda_{\min}$ equally spaced on a log-base-2 scale. We found that these choices of $\lambda_{\max}$ also seemed to work well when $n < q$, although they are not sufficient to guarantee that $\hat{\beta} = 0$ in this setting.

4 Simulation studies

4.1 Data generating models

In this section, we compare (2) to alternative methods for fitting the multivariate response linear regression model in high-dimensional settings. We consider four distinct data generating models. In Section 4.4, we also study various approaches for tuning parameter selection in (2).

For one hundred independent replications, we generated $n_{\text{train}}+n_{\text{validate}}+n_{\text{test}}$ independent realizations of $X \sim \mathrm{N}_p(0, \Sigma_{*X})$ and generated $Y \mid X = x \sim \mathrm{N}_q\left(\beta_*'x, \Omega_*^{-1}\right)$ with the structure of $\Omega_*^{-1} \equiv \Sigma_*$ differing in each model. The models we considered are:

Model 1: $\Sigma_* = D\tilde{\Sigma}_* D$, where $D \in \mathbb{R}^{q\times q}$ has diagonal entries equally spaced from 0.5 to 3.0 and zeros elsewhere; and $\tilde{\Sigma}_{*j,k} = \xi^{|j-k|}$ for $(j,k) \in \{1,\ldots,q\} \times \{1,\ldots,q\}$;

Model 2: $\Sigma_* = D\tilde{\Sigma}_* D$, where $D \in \mathbb{R}^{q\times q}$ has diagonal entries equally spaced from 0.5 to 3.0 and zeros elsewhere; and $\tilde{\Sigma}_{*j,k} = \xi \mathbf{1}(j \neq k) + \mathbf{1}(j = k)$ for $(j,k) \in \{1,\ldots,q\} \times \{1,\ldots,q\}$, where $\mathbf{1}(\cdot)$ is the indicator function.

Model 3: $\Sigma_* = D\tilde{\Sigma}_* D$, where $\in \mathbb{R}^{q\times q}$ has diagonal entries equally spaced from 0.5 to 3.0 and zeros elsewhere; and $\tilde{\Sigma}_{*j,k} = O^q_{(q)} \Gamma {O^q_{(q)}}'$ where $O^q_{(q)}$ is a randomly generated $\mathbb{R}^{q\times q}$ orthogonal matrix and Γ is diagonal with equally spaced entries from 1 to $(\mathrm{cond})^{-1}$, where (cond) is a condition number.

In addition to Models 1 – 3, we also considered a model where the errors followed a multivariate t-distribution with three degrees of freedom and covariance equivalent to that under Model 1. As before, we set $E(Y \mid X = x) = \beta_*' x$ in this setting as well. We refer to this model as *Model 4*.

The covariance used in Liu et al. (2015) is similar to that in both Model 1 and Model 2 with $\xi = 0$ (in which case Model 1 and Model 2 are equivalent). The precision matrices under Model 1 and 4 are relatively sparse, whereas under Model 2 and 3, the precision matrices can be relative dense. Under Model 4, the normality assumption made by some competing methods is violated.

Throughout our simulations, we fixed $n_{\mathrm{train}} = n_{\mathrm{validate}} = 200$, $n_{\mathrm{test}} = 1000$, $p = 500$, $q = 50$, and let ξ or the condition number vary. In the Supplementary Material, we also display results from simulations under Model 1 and 2 with $\xi = .9$ and q varying.

Independently in each replication, we generated the regression coefficient matrix $\beta_* = S \circ Q$ where $S \in \mathbb{R}^{p\times q}$, $Q \in \mathbb{R}^{p\times q}$, and $\circ$ denotes the elementwise product. The matrix S encodes the sparsity of β_*: S has between three and five randomly selected entries equal to one per column, with all other entries equal to zero. The matrix Q has entries randomly set to 1 or -1 with equal probability. Thus, the matrix β_* had an expected percentage of nonzeros equal less than 1%, but signals can be relatively strong.

4.2 Competing methods

Since β_* was sparse under our data generating models, we compared multiple sparsity-inducing estimators. The methods we compared are:

- `MSRL-Val`: The multivariate square root lasso from (2) with tuning parameter chosen by minimizing the validation set squared prediction error averaged across all q responses.
- `Calibrated`: A variation of the calibrated multivariate response linear regression method proposed by Liu et al. (2015):

$$\underset{\beta\in\mathbb{R}^{p\times q}}{\arg\min} \left[\sum_{k=1}^{q} \left\{ \frac{1}{\sqrt{n}} \|Y_{\cdot,k} - X\beta_{\cdot,k}\|_2 + \lambda \sum_{j=1}^{p} |\beta_{j,k}| \right\} \right],$$

where λ is chosen by minimizing the validation set prediction error averaged across all q responses. Note that this estimator is equivalent to q separate univariate square-root lasso estimators (Belloni et al., 2011) with the same tuning parameter λ used for each response.

- `Lasso-q`: Separate lasso estimators for each of the q columns of β_*, i.e.,

$$\underset{\beta \in \mathbb{R}^{p \times q}}{\arg\min} \sum_{k=1}^{q} \left\{ (2n)^{-1} \|Y_{\cdot,k} - X\beta_{\cdot,k}\|_2^2 + \lambda_k \sum_{j=1}^{p} |\beta_{j,k}| \right\}, \tag{17}$$

  where each λ_k is chosen by minimizing the kth response's validation set squared prediction error for $k = 1, \ldots, q$.

- `Lasso-1`: The lasso estimator with a single tuning parameter selected for all q responses, i.e., (17) where $\lambda = \lambda_1 = \lambda_2 = \cdots = \lambda_q$ with λ chosen to minimize squared prediction error on the validation set averaged across all q responses.

- `MRCE-ap`: The approximate version of the multivariate regression with covariance estimation method proposed by Rothman et al. (2010). This estimator is computed in two steps:

  1. Obtain $\beta^{(0)}$, the `Lasso-q` estimator.
  2. Set $S = n^{-1}(Y - X\beta^{(0)})'(Y - X\beta^{(0)})$ and compute

$$\Omega_\gamma^{(1)} = \underset{\Omega \in \mathbb{S}_+^q}{\arg\min} \left\{ \operatorname{tr}(S\Omega) - \log\det(\Omega) + \gamma \sum_{j,k} |\Omega_{j,k}| \right\}.$$

  Then, with $\Omega_\gamma^{(1)}$ fixed, the `MRCE-ap` estimator is

$$\underset{\beta \in \mathbb{R}^{p \times q}}{\arg\min} \left\{ n^{-1} \operatorname{tr}\left[(Y - X\beta)\Omega_\gamma^{(1)}(Y - X\beta)' \right] + \lambda \sum_{j,k} |\beta_{j,k}| \right\}. \tag{18}$$

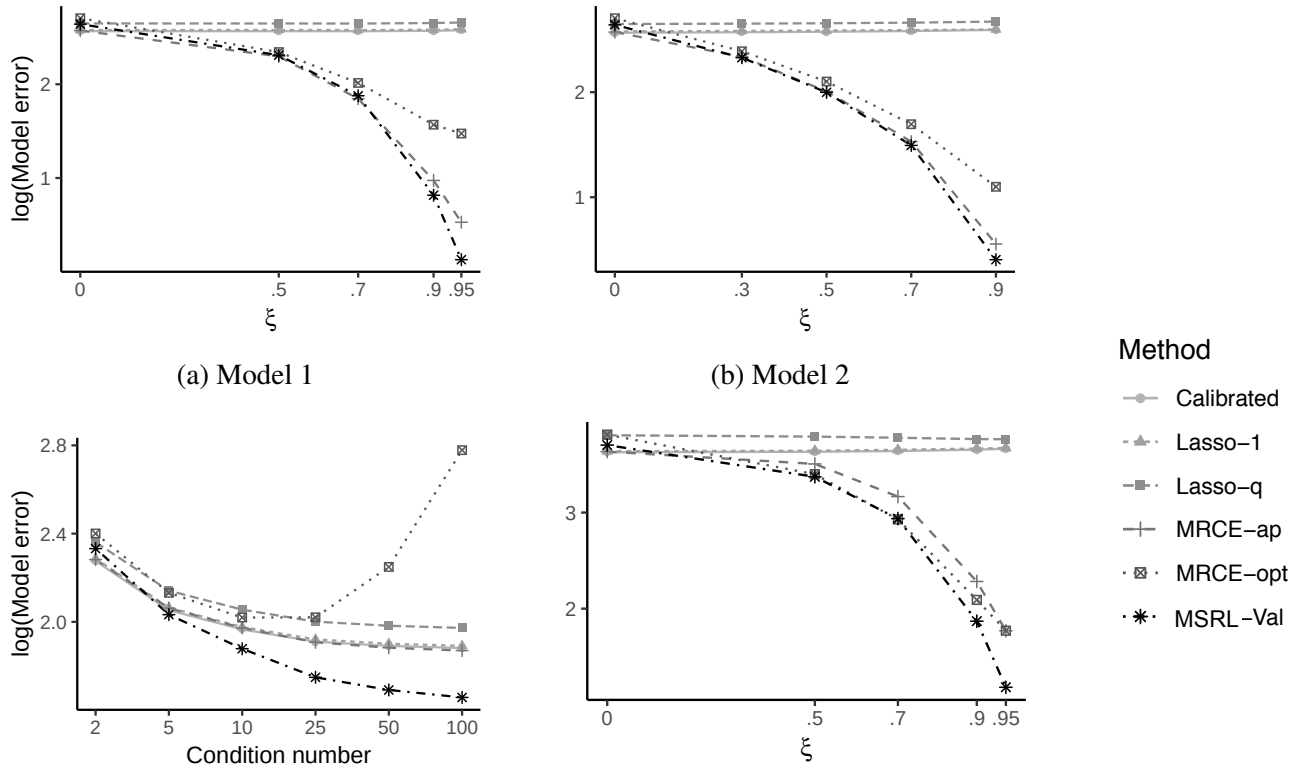
  The tuning parameter pair (γ, λ) is chosen by minimizing the validation squared prediction error averaged across all q responses.

- `MRCE-opt`: The penalized normal maximum likelihood estimator of β_* with Ω_* known, i.e., (18) with $\Omega_\gamma^{(1)}$ replaced with Ω_*. The tuning parameter λ is chosen by minimizing the validation set prediction error averaged across all q responses.

For all ξ under Model 1, we expected both `MRCE` methods to perform well since the population precision matrix is tridiagonal. The estimator `MRCE-opt` was meant to serve as the "optimal" approach for methods that jointly estimate both β_* and Ω_*. We found that the computing time for the exact version of the method

proposed by Rothman et al. (2010) could be extremely long for our data generating models, so we only compared to the optimal and approximate versions.

To measure the performance of each estimator, we followed Yuan et al. (2007) and Molstad and Rothman (2016) by using model error: $\|\Sigma_{*X}^{1/2}(\beta_* - \hat{\beta})\|_F^2$ where $\hat{\beta}$ is some estimate of the regression coefficient matrix β_*. In the Supplementary Material, we also display results for a weighted prediction error.



(a) Model 1 (b) Model 2

(c) Model 3 (d) Model 4

Figure 1: Average log-model error over one hundred independent replications under Model 1 – 4 with ξ or (cond) varying and $q = 50$.

4.3 Results

In Figure 1 we display the average model error for the six methods we considered. For both Model 1 and Model 2, when $\xi = 0$, the best performing method in terms of model error was `Calibrated`. This is not surprising since these settings conform to the model assumptions of Liu et al. (2015). Of our method and the two versions of the method of Rothman et al. (2010), `MRCE-ap` performed best when $\xi = 0$. When the error correlation was high or when Σ_* had condition number greater than five, our method outperformed all competitors in terms of model error. In general, `MSRL-Val` tended to perform similarly to `MRCE-ap`, which requires the selection of two tuning parameters and is much more computationally intensive. Under

	Model 1										Model 2									
	TPR					FPR ($\times 100$)					TPR					FPR ($\times 100$)				
ξ	0	0.5	0.7	0.9	0.95	0	0.5	0.7	0.9	0.95	0	0.3	0.5	0.7	0.9	0	0.3	0.5	0.7	0.9
Calibrated	1.00	1.00	1.00	1.00	1.00	4.80	4.85	4.86	4.86	4.89	1.00	1.00	1.00	1.00	1.00	4.80	4.81	4.81	4.85	4.87
Lasso-1	1.00	1.00	1.00	1.00	1.00	4.74	4.77	4.75	4.77	4.83	1.00	1.00	1.00	1.00	1.00	4.74	4.75	4.76	4.78	4.78
Lasso-q	1.00	1.00	1.00	1.00	1.00	4.12	4.14	4.10	4.19	4.25	1.00	1.00	1.00	1.00	1.00	4.12	4.12	4.15	4.18	4.29
MRCE-ap	1.00	1.00	1.00	1.00	1.00	4.68	5.50	5.05	4.62	4.56	1.00	1.00	1.00	1.00	1.00	4.68	5.88	5.73	5.55	5.51
MRCE-opt	1.00	1.00	1.00	1.00	1.00	5.93	5.11	3.99	2.03	1.43	1.00	1.00	1.00	1.00	1.00	5.93	5.72	5.31	4.34	2.27
MSRL-Asymp	0.70	0.76	0.84	0.95	0.98	0.00	0.00	0.00	0.00	0.00	0.70	0.79	0.88	0.95	0.99	0.00	0.00	0.00	0.00	0.00
MSRL-Asymp-2	1.00	1.00	1.00	1.00	1.00	0.82	0.84	0.85	0.88	0.88	1.00	1.00	1.00	1.00	1.00	0.82	0.82	0.82	0.82	0.82
MSRL-Val	1.00	1.00	1.00	1.00	1.00	4.54	4.47	4.44	4.43	4.52	1.00	1.00	1.00	1.00	1.00	4.54	4.55	4.56	4.59	4.68
MSRL-q95	0.89	0.93	0.97	1.00	1.00	0.00	0.00	0.00	0.00	0.00	0.89	0.95	0.98	1.00	1.00	0.00	0.00	0.00	0.00	0.00
MSRL-q95-2	1.00	1.00	1.00	1.00	1.00	1.65	1.68	1.70	1.73	1.73	1.00	1.00	1.00	1.00	1.00	1.65	1.66	1.65	1.66	1.66

Table 1: Average true positive and false positive (times 100) rates for identifying nonzero entries in β_*.

Models 1 – 3, both `MRCE-ap` and `MSRL-Val` outperformed `MRCE-opt`, the penalized maximum likelihood estimator of β_* which has oracle knowledge of Ω_*. Under Model 4, where the multivariate normality assumption of both `MRCE` variants is violated, we see `MRCE-opt` outperformed `MRCE-ap`, both of which were outperformed by `MSRL-Val` as ξ increases.

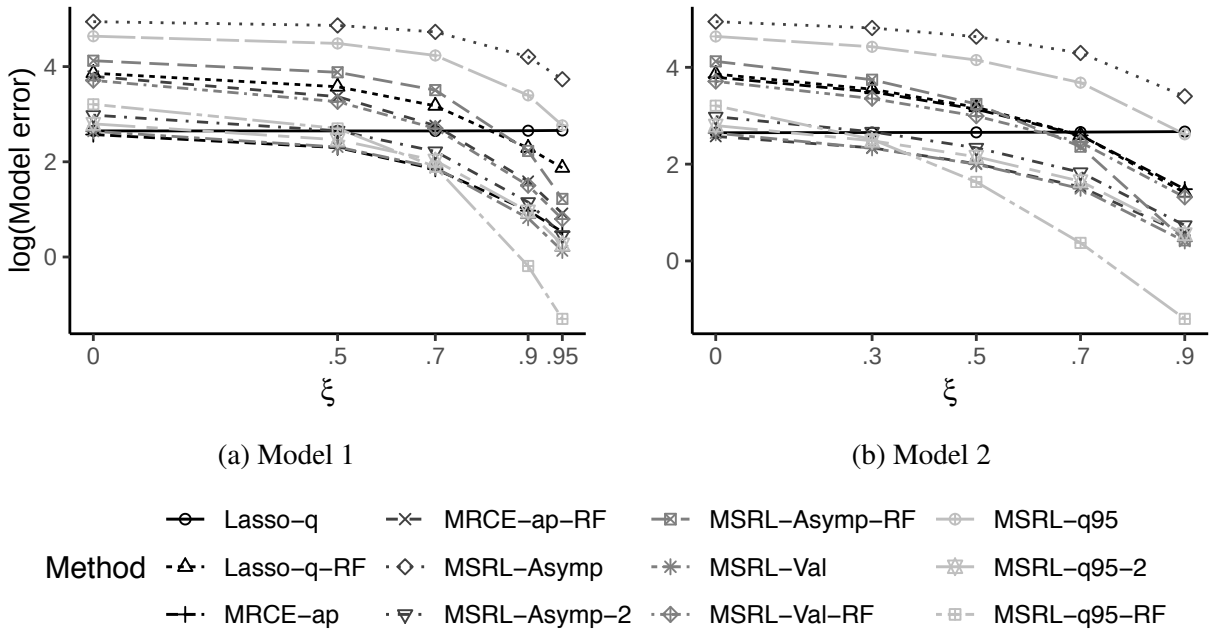
In the Supplementary Material, we display weighted prediction error for the same settings displayed in Figure 1. The relative performances were similar to those in Figure 1 – with `MSRL-Val` outperforming all competitors when the error correlations were high and when Σ_* was moderately ill-conditioned. In addition, we also display both model error and weighted prediction error under Models 1 and 2 with $\xi = 0.9$ and q taking values in $\{25, 50, 100, 150\}$. We found that as q increased, `MSRL-Val` and `MRCE-ap` tended to perform more similarly, with `MSRL-Val` outperforming `MRCE-ap` when $q = 25$, but with the two displaying similar performance when $q = 150$.

4.4 Theoretical tuning results

Based on our theoretical results, we also studied selecting tuning parameters for (2) in two more computationally efficient ways: (i) based on quantiles of the empirical distribution of the random variable $\frac{c}{\sqrt{n}}\|X'O_q^{(n)}\|_{\max}$ where O is a random matrix whose distribution is uniform over the set of $n \times q$ semi-orthogonal matrices and (ii) based on the analytic expression from Theorem 1. We used these approaches with ξ varying for Models 1 and 2. Note that these approaches require fitting (2) for only a single tuning parameter, and thus, are far more computationally efficient than the competing methods.

To approximate the distribution of the random variable used in (i), we generated 10,000 orthogonal matrices $O_q^{(n)}$ independently by generating $U \in \mathbb{R}^{n\times q}$ whose entries are iid $\mathrm{N}(0,1)$ and setting $O_q^{(n)} = U(U'U)^{-1/2}$.

We tried three versions of the two tuning approaches: `MSRL-q95` sets λ equal to the 95th quantile of the empirical distribution of $\frac{1.01}{\sqrt{n}}\|X'O_q^{(n)}\|_{\max}$. Like Belloni et al. (2011), we found `MSRL-q95` performed well at variable selection, often having a false positive rate of zero. However, like Belloni et al. (2011), we also found that this choice of λ led to substantial bias. Thus, we also used `MSRL-q95-RF`, a refitted



(a) Model 1

(b) Model 2

Figure 2: Average log-model error over one hundred independent replications under Model 1 - 2 with ξ varying.

version of `MSRL-q95` where we re-estimate the coefficients using a likelihood-based seemingly unrelated regression estimator described in the Supplementary Material. In addition to the refitted estimator, we also consider a version which the 95th quantile tuning parameter times one-half, which we call `MSRL-q95-2`. We also tried variations of `MSRL` based on the analytic expression for λ from Theorem 1. Specifically, for `MSRL-Asymp`, we use $\lambda = 1.01\left\{2n^{-1}\log(2pq/.05)\right\}^{1/2}$ and repeated the same refitting procedure to obtain `MSRL-Asymp-RF`. In addition, we use $\lambda = \frac{1.01}{2}\left\{2n^{-1}\log(2pq/.05)\right\}^{1/2}$ to obtain `MSRL-Asymp-2`.

Model error averages (on the log scale) for these approaches are displayed in Figure 2. Average true positive and false positive rates are displayed in Table 1. As observed in Belloni et al. (2011), the theory-based tuning parameters selected smaller models than validation-based tuning parameters. In particular, when $\xi > .70$, the tuning parameter based on the quantiles of $\frac{1.01}{\sqrt{n}}\|X'O_q^{(n)}\|_{\max}$ had nearly perfect variable selection accuracy, whereas the validation based approach tended to include many irrelevant predictors. However, refitting appears necessary to counterbalance the additional bias incurred from using the nuclear norm as a loss function – a conclusion also drawn by Belloni et al. (2011). Both versions of `MSRL` which use one-half times the theory-based tuning parameter values tended to perform similar to the validation-based version `MSRL-Val`. Together, these results suggest that theory-based tuning could be a useful alternative to validation-based tuning when variable selection accuracy and computational efficiency are priorities.

	Weighted prediction error				Nuclear norm prediction error			
m	20		40		20		40	
g	500	1000	500	1000	500	1000	500	1000
`MSRL-Val`	0.6424	0.6103	0.6698	0.6435	0.2128	0.2069	0.3388	0.3317
`Lasso-1`	0.6518	0.6164	0.6747	0.6442	0.2146	0.2086	0.3403	0.3329
`Lasso-q`	0.6518	0.6167	0.6764	0.6455	0.2148	0.2088	0.3422	0.3347
`MSRL`*	0.6413	0.6073	0.6690	0.6413	0.2127	0.2068	0.3387	0.3319
`MRCE-ap`*	0.6416	0.6060	0.6659	0.6354	0.2130	0.2069	0.3387	0.3314

Table 2: Weighted prediction error and nuclear norm prediction error averaged over 100 training/testing splits for the five considered methods.

5 Glimoblastoma multiforme application

We used our method to model the linear relationship between microRNA expression and gene expression in patients with glimoblastoma multiforme, a brain cancer, collected by the Cancer Genome Atlas Project (TCGA, Weinstein et al. (2013)). We were motivated to apply our method to these data as earlier versions of this dataset were analyzed by Wang (2015) and Lee and Liu (2012), both of whom proposed new methods for multivariate response linear regression which explicitly modelled the error precision matrix. Following both Wang (2015) and Lee and Liu (2012), mircoRNA expression profiles were treated as the response and gene expression profiles were treated as predictors. The preprocessed data were obtained using the `TCGA2STAT` R package (Wan et al., 2015): gene expression data was measured on an Agilent 244K Custom Gene Expression G4502A-07 microarray and microRNA were measured on an Agilent 8x15K Human miRNA-specific microarray.

Following Wang (2015), we first reduced the dimensionality of both predictors and responses, keeping the g genes with largest median absolute deviation and the m microRNAs with largest median absolute deviation. We then removed 93 subjects whose first two principal components for gene expression were substantially different than the majority of subjects: after removing these patients, there were 397 subjects in our complete dataset. An R script for creating the datasets we analyzed is available at github.com/ajmolstad/MSRL.

For one hundred independent replications, we randomly split the data into training and testing sets of size 250 and 147 respectively. We fit the multivariate response regression model using four separate methods described in Section 4.2: `MSRL-Val`, `Lasso-q`, `Lasso-1`, and a variation of `MRCE-ap`. For `MSRL-Val`, `Lasso-q`, and `Lasso-1`, tuning parameters are selected by five fold cross validation minimizing prediction error. Unfortunately, computing times for `MRCE-ap` could be extremely long, so we tried "best-case" tuning, i.e., we select the tuning parameters which gives the minimum prediction error on the test set. Note that this approach is not applicable in practice, but is included to demonstrate that `MSRL-Val` performs similarly to the much more computationally intensive approach. For comparison, we also include the "best-case" tuning version of `MSRL`. We denote both of these versions with a superscript $*$ in Table 2.

We compared the five methods in terms of two prediction metrics: weighted prediction error, $\|(Y_{\text{test}} - \hat{Y}_{\hat{\beta}})\Lambda^{-1}\|_F^2/147m$ and nuclear-norm prediction error, $\|Y_{\text{test}} - \hat{Y}_{\hat{\beta}}\|_*/1000$; where Λ is a diagonal matrix with the complete data response standard deviations along its diagonal. For `Lasso-1` and `Lasso-q`, we standardized the response variables for model fitting whereas for our method, we instead weighted the ℓ_1-norm penalty by the response standard deviations. For `MRCE-ap`*, we standardized response variables so that the precision matrix estimate was on the correlation scale.

In Table 2, we display prediction errors averaged over 100 replications in the various settings. Amongst the methods which could be used in practice, `MSRL-Val` substantially outperformed both `Lasso-1` and `Lasso-q` in terms of weighted prediction error when $g = 500$. When $g = 1000$, `MSRL-Val` performed only slightly better than either `Lasso` variant. Both "best-case" methods performed slightly better than `MSRL-Val`, with the more computationally intensive `MRCE-ap`* slightly outperforming `MSRL`* in the higher-dimensional settings. In terms of nuclear norm prediction error, `MSRL-Val` outperformed both `Lasso` variants in every setting, and even performs better than `MRCE-ap`* in some settings.

6 Discussion

In this article, we study the multivariate square-root lasso from theoretical, computational, and empirical perspectives. However, there are a number of important directions for future research. First, additional work is needed to understand the asymptotic behavior of estimates of β_* based on the refitting procedure used in Section 4. In particular, because ordinary least squares fails to account for dependent errors, the refitting procedure we employ explicitly estimates the error precision matrix. When q is large, this approach may be infeasible or may perform poorly due to ill-conditioning of the error precision matrix.

Second, the prox-linear ADMM algorithm we propose in Section 3.2 could also be applied to a broad class of penalized regression estimators which use non-differentiable loss functions, e.g., using the ℓ_1-norm as in least absolute deviation regression (Wang, 2013), or a generalization using the matrix 1-norm.

Finally, because (2) is convex and does not require an estimate of the error covariance matrix to account for dependent errors, (2) is a reasonable alternative to some existing methods, e.g., Rothman et al. (2010) or Yin and Li (2011) when an estimate of the precision matrix is not needed. However, even if an estimate of the error covariance or precision matrix is also desired by the practitioner, Remark 1 suggests that the estimate obtained by solving (2) may perform reasonably well in certain settings. Unfortunately, when q is large, there is no guarantee that this estimate will be positive definite. Hence, one may consider using (6) after introducing an additional positive definiteness constraint on the optimization variable corresponding to Σ_*.

References

Beck, A. and Teboulle, M. (2009). Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems. *IEEE Transactions on Image Processing*, 18(11):2419–2434.

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

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

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

Deng, W. and Yin, W. (2016). On the global and linear convergence of the generalized alternating direction method of multipliers. *Journal of Scientific Computing*, 66(3):889–916.

Eaton, M. L. (1989). *Chapter 7: Random orthogonal matrices*, volume 1 of *Regional Conference Series in Probability and Statistics*, pages 100–107. Institute of Mathematical Statistics and American Statistical Association, Haywood CA and Alexandria VA.

Grant, M. and Boyd, S. (2014). CVX: Matlab software for disciplined convex programming, version 2.1. http://cvxr.com/cvx.

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

Lange, K. (2016). *MM optimization algorithms*, volume 147. SIAM.

Lee, W. and Liu, Y. (2012). Simultaneous multiple response regression and inverse covariance matrix estimation via penalized Gaussian maximum likelihood. *Journal of Multivariate Analysis*, 111:241–255.

Li, X., Jiang, H., Haupt, J., Arora, R., Liu, H., Hong, M., and Zhao, T. (2016). On fast convergence of proximal algorithms for SQRT-Lasso optimization: Don't worry about its nonsmooth loss function. *arXiv preprint arXiv:1605.07950*.

Liu, H., Wang, L., and Zhao, T. (2015). Calibrated multivariate regression with application to neural semantic basis discovery. *Journal of Machine Learning Research*, 16:1579–1606.

Molstad, A. J. and Rothman, A. J. (2016). Indirect multivariate response linear regression. *Biometrika*, 103(3):595–607.

Obozinski, G., Wainwright, M. J., and Jordan, M. I. (2011). Support union recovery in high-dimensional multivariate regression. *The Annals of Statistics*, 39(1):1–47.

Parikh, N. and Boyd, S. (2014). Proximal algorithms. *Foundations and Trends® in Optimization*, 1(3):127–239.

Rothman, A. J., Levina, E., and Zhu, J. (2010). Sparse multivariate regression with covariance estimation. *Journal of Computational and Graphical Statistics*, 19(4):947–962.

Stucky, B. (2017). *Asymptotic Confidence Regions and Sharp Oracle Results under Structured Sparsity*. PhD thesis, ETH Zurich.

Sun, T. and Zhang, C.-H. (2012). Scaled sparse linear regression. *Biometrika*, 99(4):879–898.

Turlach, B. A., Venables, W. N., and Wright, S. J. (2005). Simultaneous variable selection. *Technometrics*, 47(3):349–363.

Van de Geer, S. (2016). Estimation and testing under sparsity. *Lecture Notes in Mathematics*, 2159.

Van de Geer, S. and Stucky, B. (2016). $\chi 2$-confidence sets in high-dimensional regression. In *Statistical Analysis for High-Dimensional Data*, pages 279–306. Springer.

Wan, Y.-W., Allen, G. I., and Liu, Z. (2015). TCGA2STAT: simple TCGA data access for integrated statistical analysis in R. *Bioinformatics*, 32(6):952–954.

Wang, J. (2015). Joint estimation of sparse multivariate regression and conditional graphical models. *Statistica Sinica*, 25(3):831–851.

Wang, L. (2013). The L1 penalized LAD estimator for high dimensional linear regression. *Journal of Multivariate Analysis*, 120:135 – 151.

Watson, G. A. (1992). Characterization of the subdifferential of some matrix norms. *Linear Algebra and its Applications*, 170:33–45.

Weinstein, J. N., Collisson, E. A., Mills, G. B., Shaw, K. R. M., Ozenberger, B. A., Ellrott, K., Shmulevich, I., Sander, C., Stuart, J. M., and Network, C. G. A. R. (2013). The cancer genome atlas pan-cancer analysis project. *Nature Genetics*, 45(10):1113.

Yin, J. and Li, H. (2011). A sparse conditional gaussian graphical model for analysis of genetical genomics data. *The Annals of Applied Statistics*, 5(4):2630.

Yuan, M., Ekici, A., Lu, Z., and Monteiro, R. (2007). Dimension reduction and coefficient estimation in multivariate linear regression. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 69(3):329–346.